

KARELIA-AMMATTIKORKEAKOULU

Sähkötekniikan koulutusohjelma

Jarkko Nevalainen

ILMANLAADUN SEURANTAJÄRJESTELMÄN TOTEUTTAMINEN

Opinnäytetyö
Toukokuu 2014



OPINNÄYTETYÖ
Toukokuu 2014
Sähkötekniikan koulutusohjelma

Karjalankatu 3
80200 JOENSUU
p. 013 260 6800

Tekijä
Jarkko Nevalainen

Nimeke
Ilmanlaadun seurantajärjestelmän toteuttaminen

Toimeksiantaja
Karelia-ammattikorkeakoulu

Tiivistelmä

Opinnäytetyön aiheena oli suunnitella ja rakentaa omakotitalon ilmanlaatua mittaava ja dokumentoiva järjestelmä. Lisäksi tavoitteena oli liittää laitteisto kohdetalon ilmanvaihtojärjestelmään ja säätää ilmanvaihtoa ilmanlaadusta riippuen. Järjestelmä toteutettiin Arduino-elektroniikka-alustaan sekä Xively-pilvipalveluun perustuen.

Opinnäytetyön teoreettisessa osuudessa käsiteltiin rakennusautomaatiota, sisäilman laatua ja työssä käytettyjen tekniikoiden, antureiden sekä moduulien ominaisuuksia. Teoreettisessa osuudessa perehdyttiin myös Arduino-alustoihin sekä tarkasteltiin niiden soveltuvuutta ja liitettävyyttä automaation sovelluksiin.

Toiminnallisessa osuudessa selvitettiin yksityiskohtaisesti järjestelmän toteuttamisen eri vaiheet ja käytiin läpi mahdolliset kehitys- ja laajennusmahdollisuudet. Työn tuloksena saatiin toimiva järjestelmä, joka mittaa sisä- ja poistoilmasta suhteellista kosteutta, lämpötilaa sekä sisäilman hiukkaspitoisuutta. Järjestelmä myös lähettää tiedot etäpalvelimelle ja ohjaa ilmastointilaitteen poistoilmapuhallinta kosteusarvojen säätelämänä. Järjestelmä sopii yksityiskäyttöön hyvin, mutta kehittäminen kaupalliseksi tuotteeksi vaatisi ainakin laitteen ja palvelimen välisen tietoliikenteen tietoturvan parantamista.

Kieli
suomi

Sivuja 47
Liitteet 1
Liitesivumäärä 1

Asiasanat
mikro-ohjaimet, sisäilma, rakennusautomaatio



THESIS
May 2014
Degree Programme in Electrical Engineering
Karjalankatu 3
FI 80200 JOENSUU
FINLAND
Tel. +358 (13) 260 6800

Author
Jarkko Nevalainen

Title
Development and Implementation of Air Quality Monitoring System

Commissioned by
Karelia University of Applied Sciences

Abstract

The subject of this thesis was to design and build a system monitoring indoor air quality of a detached house. It was also a priority to connect the device to the air ventilation system and control the exhaust ventilation. The system was created by using the Arduino electronics prototyping platform and Xively cloud computing services.

The theoretical section of the thesis was based on examining home automation, indoor air quality as well as the technologies, sensors and modules applied in the system. The theoretical part included also exploring the features and connectivity of Arduino based platforms as well as considering the advantages and disadvantages of applying Arduino to automation.

In the functional section of the thesis it was particularly explained how the process is implemented. Furthermore, the expansion and development opportunities were also studied and rationalized.

The result is the working apparatus measuring the temperature and relative humidity from the supply and exhaust air. The device also measures the particulate matter and sends all the measuring data to cloud server. The measuring information can be viewed in the personal Xively development page. The exhaust air fan of the air ventilation system is controlled by the values of relative humidity. The conclusion is that the system developed in this thesis is suitable for personal usage and applications but not yet in the commercial use as the data connection lacks the necessary encryption.

Language
Finnish

Pages 47
Appendices 1
Pages of Appendices 1

Keywords
microcontrollers, indoor air quality, home automation

Sisältö

Lyhenteet

1	Johdanto	7
2	Rakennusautomaatio	8
3	Internet of Things ja Xively	8
4	Sisäilman laatu	10
4.1	Ilmankosteus ja lämpötila	10
4.2	Hiukkaspäästöt	11
4.3	Ilmanlaadun parantaminen	12
5	Ilto 440 Econo	12
6	Arduino	13
6.1	Arduino Mega	14
6.2	Arduino-ohjelmisto ja ohjelmointikieli	15
6.3	Arduino ja anturit	17
6.3.1	A/D-muunnos	18
6.3.2	Pulssileveysmodulaatio	20
6.4	Arduinon soveltuvuus rakennusautomaatioon	21
6.5	Arduinon liitettävyyys automaatiojärjestelmiin	22
6.6	Arduinon tiedonsiirto lisälaitteiden kanssa	23
7	Työssä käytettävät anturit ja moduulit	24
7.1	Hiukkasanturi GP2Y1010AU0F	24
7.2	Hanrun Ethernet Shield	26
7.3	Kosteus- ja lämpötila-anturi DHT11	26
7.4	Reaaliaikakellomoduuli DS1307	27
8	Järjestelmän rakentaminen ja ohjelmointi	28
8.1	Reaaliaikakellon liittäminen ja ajastus	28
8.2	Lämpötilan ja suhteellisen kosteuden mittaaminen	29
8.3	Hiukkaspitoisuuden mittaaminen	32
8.4	LCD-näytön liittäminen	34
8.5	Liittyminen Ethernet-verkkoon	35
8.6	Kommunikointi Xively-palvelimen kanssa	36
8.7	Ilmastoinnin säätö	37
8.8	Kotelointi	40
9	Testaus	41
10	Laitteiston laajennusmahdollisuudet	42
11	Kustannukset	42
12	Pohdinta ja johtopäätökset	43

Lähteet

Liite Laitteen piirikaavio

Lyhenteet

A/D	Analogia/Digitaali
AI	Analog Input, Analoginen sisääntulo
AO	Analog Output, Analoginen ulostulo
DC	Direct Current, Tasavirta
DI	Digital Input, Digitaalinen sisääntulo
DO	Digital Output, Digitaalinen ulostulo
EEPROM	Electronically Erasable Programmable Read-Only Memory, Haihtumaton puolijohdemuisti, joka voidaan uudelleenkirjoittaa. Käytetään pääasiassa asetustietojen tallentamisessa mikroprosessorin sisältävissä laitteissa.
I ² C	Inter-Integrated Circuit, Kaksisuuntainen sarjamuotoinen ohjaus- ja tiedonsiirtoväylä. Tyypillinen käyttö kulutuselektroniikan vähäistä tiedonsiirtoa vaativissa sovelluksissa.
I/O	Input/Output, Sisään/Ulostulo
LCD	Liquid Crystal Display, Nestekidenäyttö
PWM	Pulse Width Modulation, Pulssinleveysmodulaatio
RH	Relative Humidity, Suhteellinen kosteus ilmoittaa prosentuaalisesti ilmaan sitoutuneen vesimäärän verrattuna kyseisessä lämpötilassa maksimaalisen määrän kosteutta sisältävään ilmaan.

SPI	Serial Peripheral Interface, Synkronoitu sarjaliikenneväylä
SRAM	Static Random Access Memory, Staattinen haihtuva puolijohde-muisti.
UART	Universal Asynchronous Receiver Transmitter, Sarjaliikennepiiri, joka muuntaa rinnakkaismuotoista tietoa sarjamuotoiseksi sekä päinvastoin.
USB	Universal Serial Bus, Sarjaväyläarkkitehtuuri oheislaitteiden liittämiseksi tietokoneeseen.

1 Johdanto

Opinnäytetyön toimeksiantajana toimi Karelia-ammattikorkeakoulu. Opinnäytetyössä rakennettiin etäluettava ilmanlaadun seurantarjestelmä. Idea opinnäytetyön aiheeseen kehittyi pitkän ajanjakson aikana. Lopullinen päätös rakentaa ilmanlaatua seuraava järjestelmä varmistui, kun kohdetalossa huomattiin olevan ongelmia ilmanvaihdon kanssa. Sisäilmaongelmiin kiinnitetään nykyisin huomiota enemmän kuin koskaan, eikä syyttä. Teollistuneessa maailmassamme erilaiset ilmansaasteet aiheuttavat terveysongelmia suurelle osalle ihmisistä. Aihe kiinnostaa läheisesti niin terveyden, kuin kiinteistön kunnonkin kannalta.

Kohteena oleva omakotitalo on 2000-luvulla rakennettu ja suhteellisen moderni varustukseltaan. Kohdetalon ilmastointikone on kuitenkin varustettu vain manuaalisella säätöyksiköllä ja yhdellä käyttöpaneelilla. Ilmanvaihdon automatisointi olikin hyödyllinen ja mielenkiintoinen lisä järjestelmän toimintaan.

Tavoitteina järjestelmälle on säästää ilmastointiin kuluva energiaa, parantaa ilmanlaatua, sekä mahdollistaa paikallinen ja etäluettava ilmanlaadun olosuhde-seuranta. Järjestelmän pohjana käytettiin Arduino Mega 2560 R3-kehitysalustaa. Piirikortille liitettiin erilaisia liitäntäyksiköitä, joiden avulla kaikki tarpeelliset mittaukset, tiedonsiirto ja muut toiminnot saatiin suoritettua. Laitteisto mittaa kiinteistön huone- ja poistoilman lämpötilaa, suhteellista kosteutta sekä sisäilman hiukkaspitoisuutta. Järjestelmä ilmoittaa mitatut arvot näytöllä, antaa tarpeen mukaan hälytyksiä, ja siirtää tiedot Xivelyn palvelimelle, josta ne ovat noudettavissa selainpohjaiseen käyttöliittymään näkyville. Laitteisto on myös liitetty Ilto 440 Econo -ilmastointilaitteeseen ja ohjaa sen poistopuhaltimen pyörimisnopeutta mittausarvojen perusteella.

2 Rakennusautomaatio

Rakennusautomaatiolla tarkoitetaan yksinkertaistettuna rakennusten teknisten laitteiden ohjaamista. Rakennusautomaatiojärjestelmä on rakennusten sisäilmastoon, valaistukseen ja turvallisuuteen vaikuttava apuväline. Sen avulla halutaan minimoida laitteiden kuluminen, energiankulutus, melu ja muut laitteiden käytöstä aiheutuvat haitat. Tyypillisiä rakennusautomaation ominaisuuksia ovat olosuhdeseuranta, mittaukset, ohjaukset, säädöt sekä toimintojen ajastukset. Väylätekniikat ovat myös vahvasti rakennusautomaation osana ja tämä mahdollistaakin monien toimintojen etäkäytön. [23.]

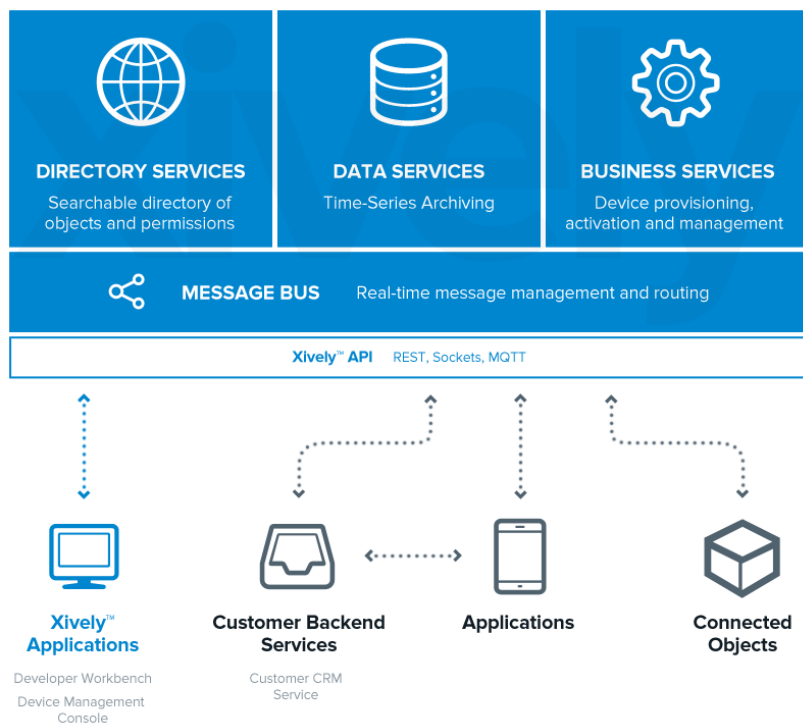
Rakennusautomaation avulla pyritään saavuttamaan kustannussäästöjä eri muodoissa. Tyypillisimpiä automatisoinnin hyötyjä ovat energiansäästö ja hallinta, parempi sisäilmasto, huolto- ja kunnossapitotoiminnan tehostaminen ja lisääntynyt turvallisuus. Rakennusautomaation avulla voidaan myös keskittää valvontaa, integroida tietojärjestelmiä ja parantaa ylläpitoa. Rakennusautomaatiojärjestelmät ovat yleistymässä vauhdilla. Järjestelmien valmistajia ja tekniikoita on myös olemassa useita. [23.]

3 Internet of Things ja Xively

Internet of Things (IoT) tarkoittaa laitteiden ja erilaisten ympäristöjen verkottunutta kommunikointia internetin välityksellä. Periaatteessa yksinkertaisimmillaan IoT on esimerkiksi lämpötila-anturin mittausdatan julkaiseminen pilvipalvelussa. Asioiden internetin on ennustettu olevan tärkeä osa tulevaisuuden tietoverkkoa. Yleisimpiä IoT:n sovelluksia ovat nykyisin esimerkiksi valaistuksen, ilmastoinnin ja lämmityksen automatisointi. Elektroniikkateollisuuden kehittyessä komponentit ovat edullisempia, pienempiä ja tehokkaampia kuin koskaan aikaisemmin. Näiden lisäksi myös verkkotekniikat ja energiatehokkuus paranevat. Yhdessä näiden muutosten ansiosta IoT on tullut yhä useampien saataville. [12, s. 8–11.]

Nämä internetin ”asiat” voivat tosin olla myös suurempia kokonaisuuksia kuten robotteja, autoja ja jopa taloja. Yksi merkittävä ero perinteiseen automaatioon verrattuna on ohjelmoinnin siirtäminen pilvipalveluihin. Eräät palvelut tarjoavat mahdollisuutta ohjata järjestelmiä internetin välityksellä käyttäjän asettamien triggereiden ohjaamina. Triggerit ovat olosuhteita tai arvoja, jotka laukaisevat käyntiin erilaisia toimintoja. Triggereinä voidaan käyttää vaikkapa paikallisia mitaustietoja tai muualta verkosta noudettuja muuttujia. [12, s. 8–11.]

Xively on yksi suosituimmista IoT-palveluista. Xively tarjoaa muun muassa pilvipalveluita, kehittäjän työkaluja, tietokantaosaamista ja sovelluksia. Xively develop on kehittäjille ilmainen, mutta lisenssit kaupalliseen käyttöön ovat maksullisia. [26.] Xively Cloud Services™ IoT-palvelun rakenne on esitetty kuvassa 1.



Kuva 1. Xively Cloud Services™ -toiminnan periaate [26]

4 Sisäilman laatu

Nykyisin sisätiloissa vietetään jopa 90 prosenttia ajasta. Tavallisesti ihminen hengittää vuorokauden aikana noin 15000 litraa ilmaa, ja sisäilman laadulla onkin suuri merkitys terveyteemme. Huonon sisäilman seurauksena työskentelemme ja viihdymme huonommin. Pitkällä aikavälillä huono sisäilma aiheuttaa erilaisia terveyshaittoja ja saattaa jopa sairastuttaa vakavasti. [2, s. 3.]

Ilmanlaatu on käsitteenä moniulotteinen. Sisäilman laatuun vaikuttavat epäsuorasti rakennuksen rakenteelliset seikat, sijainti, rakennustapa, rakennusmateriaalit, ilmanvaihtoratkaisut, käyttö ja sääolot. Ilmanlaatua huonontavia fysikaalisia tekijöitä ovat muun muassa lämpötila, melu, veto, kosteus ja radon. [4, s. 35.]

Ilmanlaatuun vaikuttavat myös ilmassa olevat kaasumaiset ja hiukkasmaiset epäpuhtaudet. Kaasumaisista epäpuhtauksista merkittäviä yhdisteitä ovat hiilidioksidi, hiilimonoksidi, rikkiyhdisteet, typen oksidit, ammoniakki ja otsoni. Sisäilman laatuun huomattavasti vaikuttavia hiukkasmaisia tekijöitä ovat biologista alkuperää olevat hiukkaset kuten sienet, virukset, bakteerit sekä kasvi- ja eläinpölyt. [4, s. 40–46.]

4.1 Ilmankosteus ja lämpötila

Lämpötila on helpoimmin aistittavissa oleva ilmanlaadun komponentti. Sisäilmastoluokka S1:n määrittämä sisäilman lämpötila on talvella 21–22 °C ja kesällä 23–24 °C. Alhaisissa pintalämpötiloissa kosteus tiivistyy kylmiin pintoihin ja ilma voi tuntua vetoisalta. Sisäilman lämpötilaan voidaan vaikuttaa lämmityksen säädöllä. [1.]

Ilman kosteudella on oma merkityksensä sisäilmaongelmien syntyyn. Liian kuiva sisäilma kuivattaa limakalvoja sekä ihoa. Toisaalta liian kostea ilma on vaaraksi talon rakenteille. Erilaiset homeet ja mikrobit tarvitsevat kasvaakseen läm-

pöä ja kosteutta. Yli 45 %:n suhteellinen kosteus parantaa huomattavasti pölypukkien mahdollisuuksia lisääntyä ja 70–80 % kosteudessa viihtyvät homesienetkin. Pitkäaikainen yli 90 %:n suhteellinen kosteus aiheuttaa kaiken lisäksi myös lahovaurioita. [1.]

Ilmankosteus kasvaa erityisesti peseytymistilojen käytöstä. Kosteutta syntyy peseytymisestä, ruuanlaitosta ja pyykkien kuivauksesta, mutta myös ihmisen ruumiintoiminnot kuten uloshengittäminen ja hikoilu tuottavat kosteutta ilmaan. Ilmassa oleva ylimääräinen kosteus tiivistyy ensimmäisenä sisäilmaa kylmemmille pinnoille, ja tämän takia olisikin viisainta yrittää pitää rakenteiden lämpötila lähellä huonelämpötilaa. Talon rakenteiden lämpötilaan ei kuitenkaan voida vaikuttaa kovinkaan helposti, ja varsinkin talvella kosteuden tiivistymisestä voi tulla ongelmia. Tällöin ainoaksi vaihtoehdoksi jääkin kosteuden poistaminen ilmanvaihdon tehostamisen avulla. [28.]

Sisäilman suhteellisen kosteuden tulisi olla talvisin 25–45 % ja kesäisin 30–60 %. Kesäisin ilmankosteus saattaa kasvaa hyvinkin suureksi sääolosuhteista johtuen. Talvisin koneellisella ilmanvaihdolla varustetuissa kiinteistöissä suhteellinen kosteus saattaa laskea vain muutamaa prosenttiin. [4, s. 37.]

4.2 Hiukkaspäästöt

Hengitysilman pienhiukkaset ($<2,5 \mu\text{m}$) ja karkeammat hengitettävät hiukkaset ($2,5\text{--}10 \mu\text{m}$) aiheuttavat lyhyellä altistusajalla hengitystieinfektioita sekä pahentavat hengityssairauksien oireita. Pienhiukkaspäästöille altistuminen pitkällä aikavälillä lyhentää elinikää, aiheuttaa keuhkohtaumatautia ja kalkkeuttaa verisuonia. [8, s. 15.] Ihmisen aiheuttamat ulkoilman hiukkaspäästöt syntyvät pääosin liikenteestä, teollisuudesta ja polttoprosesseista. [8, s. 8.]

Sisäilmaan hiukkaspäästöjä kulkeutuu ulkoilmasta, mutta päästöjä syntyy myös sisällä. Esimerkiksi kynttilöiden polttaminen, tupakointi, ruoanlaitto ja puunpoltto tuottavat sisäilmaan pienhiukkasia. Karkeampia hengitettäviä hiukkasia ilmaan siirtyy esimerkiksi tekstiileistä ja kotieläimistä. Oikeanlaisella ilmanvaihdolla on

suuri osuus hiukkasten vähentämisessä. [9.] Suomen rakentamismääräyskoelman mukaan hengitettävien hiukkasten määrä on suunnitellun ohjearvon mukaisesti korkeintaan $50 \mu\text{g}/\text{m}^3$ [20].

4.3 Ilmanlaadun parantaminen

Ilmanlaatua voidaan parantaa monella eri tavalla. Kaupalliset ilmanpuhdistajat hyödyntävät yleensä useita eri puhdistustekniikoita koska yksittäisillä tekniikoilla ei pystytä poistamaan kaikkia epäpuhtauksia. Ilmanpuhdistajat käyttävät puhdistamisessa muun muassa otsonointia, fotokatalyyttistä oksidaatiota, erilaisia suodatuksia ja ionisointia. [13.]

Käytännössä kokonaisvaltaisin helposti toteutettava tapa parantaa sisäilman laatua on tarkkailla siihen vaikuttavia mitattavissa olevia suureita sekä pitää huolta riittävästä ilmanvaihdosta. Järkevästi toteutettu ja hyvin huollettu tehokas ilmanvaihto kykenee poistamaan pieniä määriä monenlaisia epäpuhtauksia, joista muuten olisi haittaa sisäilman laadulle. Tutkimuksissa on havaittu ilmanvaihdon lisäämisellä olevan selvä vaikutus ilmanlaadun parantamisessa. Tutkimuksien mukaan ilmanvaihdon tehostaminen kymmeneen litraan sekunnissa henkilöä kohden parantaa sisäilman laatua huomattavasti. [4, s. 33–34.]

5 Ilto 440 Econo

Ilto 440 Econo on Swegon ILTO Oy:n valmistama alle 220 m^2 asuntojen ilmanvaihtoon tarkoitettu energiatehokas ilmastointilaitte. Pientalojen lisäksi sitä voidaan käyttää kerrostalojen hajautetussa ilmanvaihtojärjestelmässä. Ilmanvaihtokone on varustettu lämmön talteenottoyksiköllä ja vesikiertoisella tuloilman lämmityksellä. [25.]

Ilmastointilaitteiden markkinoijat tarjoavat usein lisälaitteita ja myös Ilto 440 -laitteelle on saatavissa monenlaisia ratkaisuja. Ilmastoinnin automaattisen ohja-

uksen avuksi on saatavilla muun muassa hiilidioksidilähetin, kosteuskytkin ja liiketunnistin. Lisälaitteet ovat kuitenkin melko kalliita ja jälkiasennettuina anturi-
en hinta on vielä korkeampi. Kohteen ilmastointilaitteelle ei myöskään ole ole-
massa kaupallista ilmanlaadun seurantayksikköä, joten haluttuja ominaisuuksia
ei edes pystyisi saavuttamaan ilman kolmannen osapuolen laitteistoa. Vakio-
ominaisuutena Ilto 440-laitteissa ilmanlaadun parantamiseksi ovat tuloilman
karkea- ja hienosuodattimet. [25.]

6 Arduino

Arduino on helppokäyttöinen vapaaseen lähdekoodiin perustuva elektroniikan
kehitysalusta. Arduino-elektroniikkakortteja ohjelmoidaan C-ohjelmointikieleen
perustuvan ohjelmointikielen avulla. Arduino-alustan ja siihen liitettävien antu-
reiden avulla voidaan rakentaa lukemattomia erilaisia elektroniikan sovelluksia.
Arduino-yhteisö toimii maailmanlaajuisesti ja apua projekteihin on saatavilla
monella eri sivustolla. [7.]

Arduino-piirikortteja on kymmeniä erilaisia, mutta muutama malli on noussut
suosituimmiksi käyttäjien keskuudessa. Suosituimpia malleja Arduino-perheestä
ovat Leonardo, Mega, Nano, Uno, Fio, Mini ja Lilypad. Kaikkia eri Arduino-
perheen malleja pystytään ohjelmoimaan samaa ohjelmistoa käyttäen. [12, s.
68–69.]

Arduino-malliston eri alustat perustuvat suurimmaksi osin samoihin komponent-
teihin, mutta voivat erota ominaisuuksiltaan laajastikin (taulukot 1 ja 2).

Taulukko 1. Arduino-alustojen yleisiä- ja muistiominaisuuksia

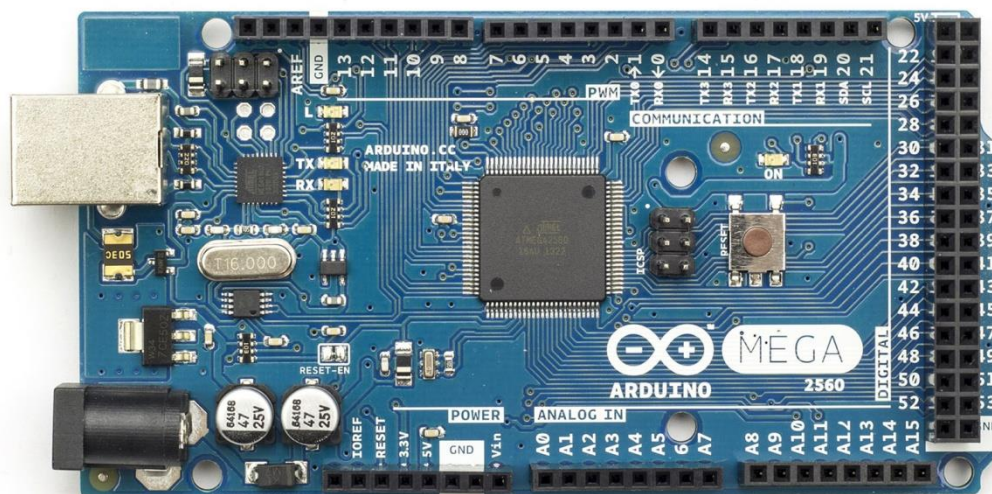
	Proessori	Käyttöjännite	Kellotaajuus	EEP- ROM	SRAM	Flash
Due	AT91SAM3X8E	3,3 V/7–12 V	84 MHz	–	96 kt	512 kt
Ethernet	Atmega 328	5 V/7–12 V	16 MHz	1 kt	2 kt	32 kt
Fio	Atmega 328P	3,3 V/3,7–7 V	8 MHz	1 kt	2 kt	32 kt
Leonardo	Atmega 32u4	5 V/7–12 V	16 MHz	1 kt	2,5 kt	32 kt
Lilypad	Atmega 168V/328V	2,7–5 V	8 MHz	0,512 kt	1 kt	16 kt
Mega2560	Atmega 2560	5 V/7–12 V	16 MHz	4 kt	8 kt	256 kt
Mini	Atmega 328	5 V/7–9 V	16 MHz	1 kt	2 kt	32 kt
Nano	Atmega 168/328	5 V/7–9 V	16 MHz	0,512–1 kt	1/2 kt	16/32 kt
Uno	Atmega 328	5 V/7–12 V	16 MHz	1 kt	2 kt	32 kt

Taulukko 2. Arduino-alustojen liitäntäominaisuuksia

	Analog Input	Analog Output	Digital I/O	PWM	UART
Due	12	2	54	12	4
Ethernet	6	–	14	4	–
Fio	8	–	14	6	1
Leonardo	12	–	20	7	1
Lilypad	6	–	14	6	–
Mega2560	16	–	54	15	4
Mini	8	–	14	6	–
Nano	8	–	14	6	1
Uno	6	–	14	6	1

6.1 Arduino Mega

Projektin toteutuksen pohjaksi valittiin Arduino Mega 2560 R3 -elektroniikkakortti (kuva 2). Valinnassa kiinnitettiin huomiota kustannuksiin, analogi- ja digitaalitulojen tarpeeseen sekä käytettävissä olevien muistien kokoihin. Tärkeimmiksi syiksi juuri Megan valitsemiseen nousivat piirikortin Flash-muistin verrattain suuri koko ja käytettävissä olevien liitäntäporttien määrä.

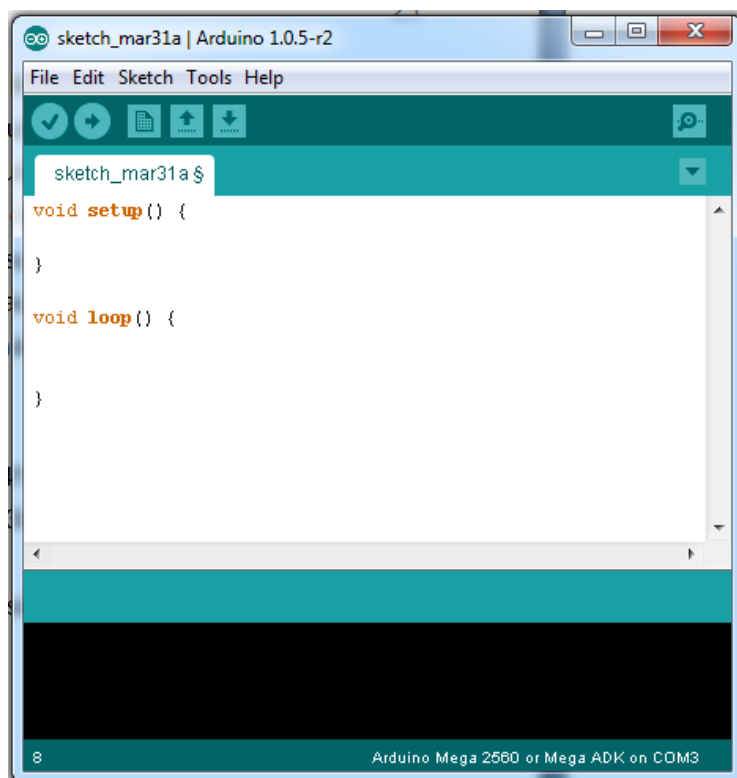


Kuva 2. Arduino Mega 2560 R3 [7]

Arduino Mega 2560 perustuu ATmega 2560 -mikroprosessoriin, joka on kelloitettu toimimaan 16 Mhz taajuudella. Piirilevyllä on 54 digitaalista I/O-porttia, 4 UART-sarjaporttia ja 16 analogista sisääntuloa. Kortin 15:tä digitaalista I/O porttia pystytään myös tarvittaessa käyttämään PWM-ulostuloina. Ohjelmointi tapahtuu USB-sarjaliikenteen avulla ja virransyöttö tapahtuu joko USB:n, pariston tai erillisen muuntajan avulla. [7.]

6.2 Arduino-ohjelmisto ja ohjelmointikieli

Arduino-korttien ohjelmointiin käytetään yleensä Arduino-sovelluskehittäjien luomaa ohjelmointiympäristöä. Arduino-ohjelmisto on saatavilla Windows, Linux ja Mac Os X -käyttöjärjestelmille. Ennen ohjelmiston käyttämistä asennetaan ohjelmoitavan piirikortin ajurit, jonka jälkeen valitaan käytettävissä oleva tietokoneen portti ohjelmointia varten. [7.] Kuvassa 3 on esitetty ohjelmointi-ikkunan perusnäkö.

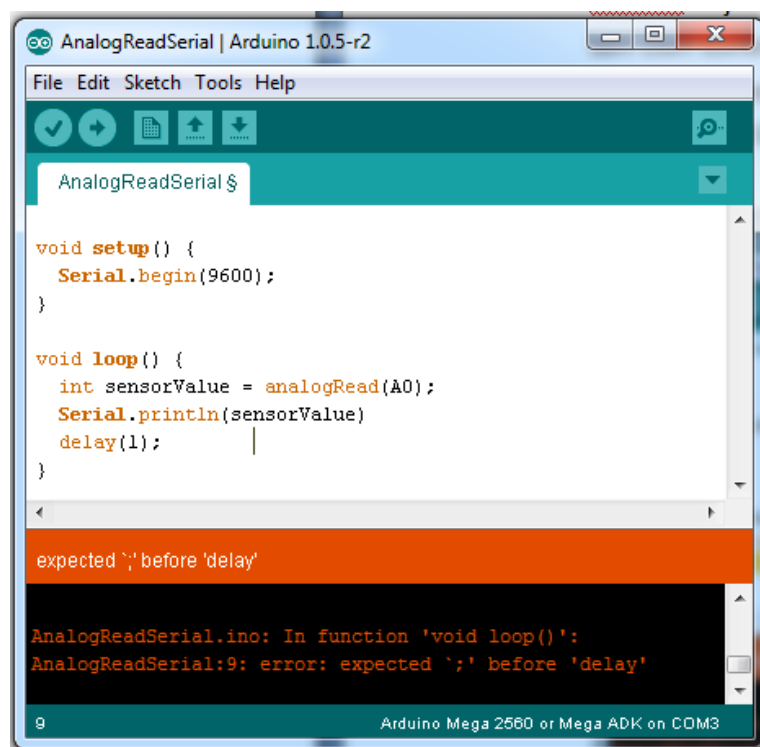


Kuva 3. Arduino-ohjelmisto v.1.05 r-2

Arduino-kortteja voidaan ohjelmoida monilla eri ohjelmointikielillä, mutta yleisimmin käytetään C-kieleen pohjautuvaa Arduino-ohjelmointikieltä. Se koostuu kolmesta ohjelmallisesta kokonaisuudesta: rakenteesta, muuttujista ja funktioista. [7.]

Rakenne sisältää ohjelmointikielen peruskomennot kuten if, if-else, while, for ja matemaattiset laskutoimitukset. Muuttujaosuudessa määritellään ohjelmassa tarvittavat muuttujat ja niiden sijainti ohjelmamuistissa. Muuttujat voidaan määrittellä tietyn pituisiksi ja muotoisiksi esimerkiksi int, word, dword ja float() komentojen avulla. Ohjelman funktio-osuus sisältää erilaisia luku- ja kirjoitusoperaatioita sekä I/O:n ohjelmalliset määrittelyt. Funktio-osuudessa voidaan suorittaa myös aikaan ja kehittyneeseen matematiikkaan perustuvia ohjelmallisia toimintoja. Funktio-osuuden komentoja ovat mm. digitalWrite(), analogRead() ja delay() toiminnot. Toimivan ohjelman täytyy rakenteellisesti sisältää vähintään void setup() ja void loop () -funktioita. [7.]

Arduino-ohjelmisto antaa palautetta ohjelman rakenteesta. Jos ohjelma on jollain osin toimimaton, niin ohjelmointiympäristö ilmoittaa viallisen koodin rivinumeron ja virhesanoman. Esimerkissä (kuva 4) on unohdettu puolipiste käskyn lopusta rivillä 9.



Kuva 4. Ohjelmoinnin virheilmoitus

6.3 Arduino ja anturit

Anturit ovat mittalaitteiden osia, joiden avulla pystytään mittaamaan fysikaalisia suureita kuten kosteus, lämpötila ja valaistusvoimakkuus. Mitattavan fysikaalisen suureen muutos aiheuttaa tyypillisesti anturista saatavan signaalin muutoksen, joka pystytään tulkitsemaan mittalaitteella. Nykyaikaiset elektroniset laitteet sisältävät runsaasti erilaisia antureita. Anturit voidaan jakaa karkeasti kahteen eri tyyppiin: analogisiin ja digitaalisiin antureihin. [6.]

Anturit voivat sisältää anturielementin lisäksi myös muita komponentteja kuten A/D-muuntimen tai vahvistimen. Mittalaite muuntaa anturilta saatavan tiedon

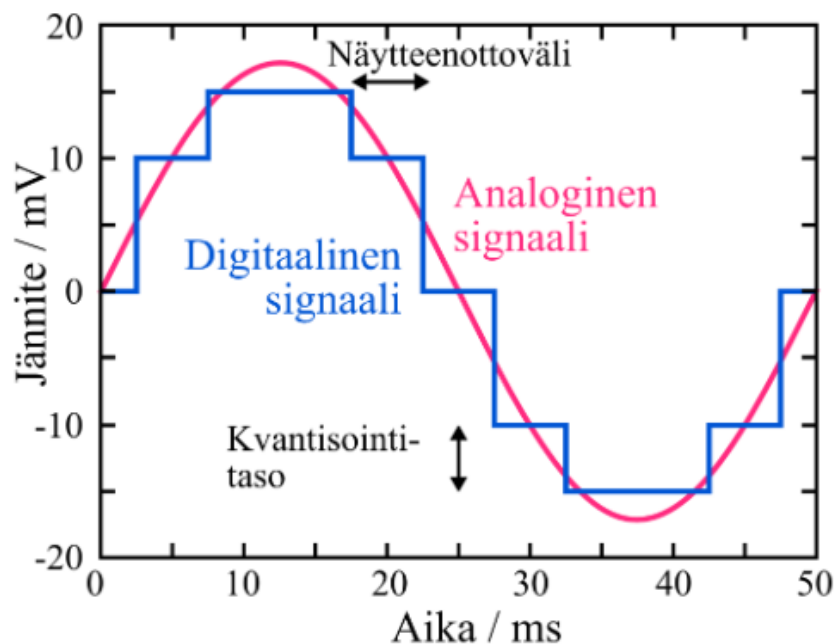
haluttuun muotoon ja ilmoittaa sen esimerkiksi näyttöyksiköllä. Tyypillisiä esimerkkejä mittalaitteista ovat vaikkapa kuumemittari ja henkilövaaka. [6.]

Analogiset anturit ovat normaalisti suunniteltu käyttäytymään vastuksen tavoin mitattavien suureiden muuttuessa. Antureista saatavan jännitteen muutos havaitaan mittalaitteen piirikortin analogisessa sisääntulossa, jonka jälkeen jännitteen arvo muutetaan ohjelmallisesti haluttuun muotoon. Tyypillisimpiä analogisia antureita ovat termistorit, valovastukset, sekä erilaiset pietsosähköiset ja kapasitiiviset anturit. [12, s. 31–37.]

Digitaalisilla antureilla on yleensä binäärinen ulostulo ja niiden signaalilähtö liitetään piirikortille digitaalisiin sisääntuloportteihin. Yksinkertaisin mahdollinen digitaalinen anturi on painonappi, jonka avulla saadaan kaksi erilaista tilaa: 0 ja 1 eli poissa tai päällä. Digitaalisien antureiden ulostulo ei ole jatkuvaa toisin kuin analoginen signaali, vaan se on eroteltu esimerkiksi yhden bitin tai tavun levyisiin sanomiin. Digitaaliset signaalit ovat tarkempia kuin analogiset sillä ne ovat vähemmän herkkiä häiriöille ja niiden arvot pystytään mittaamaan sekä erottelemaan suuremmilla taajuuksilla. Digitaalisen signaalin tulkitsemiseen vaaditaan ohjelmallisesti suoritettava muunnos. [12, s. 26–27.]

6.3.1 A/D-muunnos

A/D-muunnoksessa analoginen signaali muunnetaan digitaaliseksi. Muunnoksesta saatu koodi voidaan esittää esimerkiksi binäärisenä kahden jännitetason muutoksina [15]. A/D-muunnoksen periaatteellinen toteutus nähdään kuvassa 5.



Kuva 5. A/D-muunnoksen periaate [6, s. 15.]

Arduino Megan sisäänrakennettu A/D-muunnin muuntaa analogiasignaalin (0–5 V) 10-bittiseksi digitaalisignaaliksi, jossa minimijännite 0 V saa lukuarvon 0 ja maksimijännite 5 V lukuarvon 1023. Kymmenen bitin levyisellä signaalilla saavutetaan $2^{10}-1=1023$ erilaista jännitetasoa vastaavaa lukuarvoa.

Referenssijännitteellä 5 V ja 10 bitin resoluutiolla erottelukyky lasketaan kaavalla 1.

$$U_e = \frac{\Delta U}{2^n} = \frac{5 \text{ V}}{(2^{10}-1)} = 4,888 \text{ mV} \quad (1)$$

missä U_e = erottelukyky
 ΔU = jännitteen muutos

Useimpiin sovelluksiin tämän tasoinen erottelukyky on riittävä. Jotkin anturit tosin antavat millivolttitason jänniteviestejä eikä sellaisia pystytä tarkasti havaitsemaan ilman signaalin muokkausta.

A/D-muunnos koostuu neljästä vaiheesta:

- suodatus: suurtaajuiset häiriöt suodatetaan analogiasignaalista

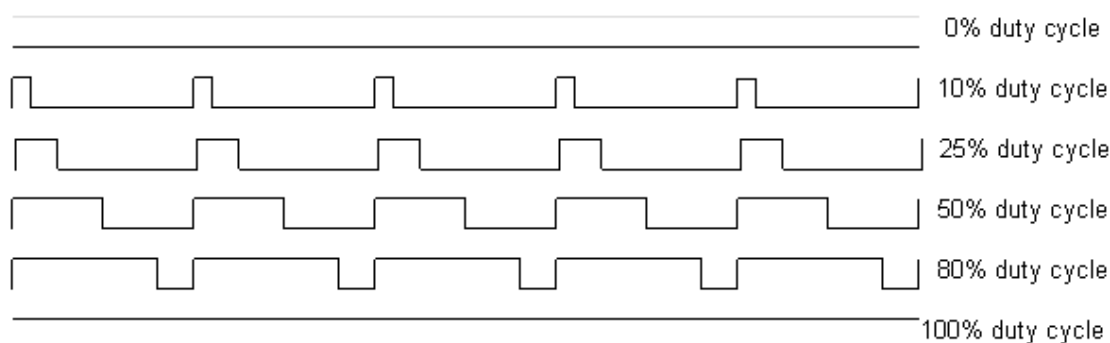
- näytteistys: signaalista otetaan tietyin aikavälein näytteitä ja tietyllä ajan hetkellä tarkastetaan näytteen tarkka arvo
- kvantisointi: näytteen jännitearvo pyöristetään lähimpään taulukoituun arvoon
- koodaus: saadut näytteet koodataan sopivaan muotoon, jonka jälkeen ne voidaan siirtää tai tallentaa laitteelle ymmärrettävästi [15.]

6.3.2 Pulssileveysmodulaatio

Edullisissa mikroprosessoriohjatuissa sulautetuissa järjestelmissä ei usein ole analogialähtöjä. Arduino-laitteista ainoastaan Due-mallissa on vakiona kaksi 0–5 V analogialähtöä ja muut mallit joutuvatkin turvautumaan pulssileveysmodulaatioon jännitesignaalin arvon muuttamisessa.

Pulssileveysmodulaatio (PWM, Pulse-Width Modulation) on signaalin modulointitapa, jonka avulla säädetään kuorman jännitteen arvoa [10, s.154]. PWM-sovelluksia käytetään esimerkiksi Led:ien himmentämiseen, DC-moottorien ohjaukseen ja analogiaviestin luomiseen. Tiettyihin sovelluksiin pelkkä PWM-lähtö on riittävä tapa moduloida jännitettä, mutta esimerkiksi oikean analogiaviestin luomiseen tarvitaan pulssileveysmodulaation lisäksi alipäästösuodinta. [21.]

Pulssileveysmodulaatiossa signaalin taajuus ($f=1/T$) pidetään yleensä vakiona ja jännitteen arvoa säädetäänkin muuttamalla pulssisuhdetta ($T1/T$). Pulssisuhde ilmaisee lähdön 1-tilassa olevan ajan suhteessa tarkasteltavan ajanjakson pituuteen. Pulssisuhde on suoraan verrannollinen jännitteen arvoon siten, että esimerkiksi 80 % pulssisuhde vastaisi $0,8 * 5 \text{ V} = 4 \text{ V}$ jännitearvoa. [10, s. 154.] Kuvasta 6 nähdään signaalin vaikutus pulssisuhteeseen.



Kuva 6. Pulssileveysmodulaation periaate [18]

Pulssileveysmodulaatiossa erottelukyky määritellään samoin, kuten A/D-muunnoksessa (kaava 1). Yhtä PWM-ulostuloa käytettäessä Atmega 2560 prosessori tarjoaa 8-bittisen resoluution, joka tarkoittaa 255 jännitetasoa [7].

PWM-ulostuloa pystytään säätämään 19,608 mV:n tarkkuudella. Yhtä ulostuloa käytettäessä ei täten pystytä kovinkaan tarkkaan jännitteen modulointiin. Synkronoinnin ja kahden ulostulon avulla on kuitenkin mahdollista suorittaa tarkempaa 16 bitin resoluutioon kykenevää pulssileveysmodulaatiota [18].

6.4 Arduinon soveltuvuus rakennusautomaatioon

Arduino-alustoja käytetään laajasti kiinteistöautomaatiossa erilaisissa harrasteprojekteissa. Rakennusautomaatiota varten rakennetut projektit ovat usein tarkoitettu lämmityksen, ilmastoinnin ja valaistuksen ohjausta varten. Esimerkkejä rakennusautomaation Arduinoon pohjautuvista ratkaisuista löytyy internetistä valtavasti. Arduino-korttien pohjalle rakennettavat järjestelmät kiinnostavat harrastajia erityisesti niiden avoimuuden, muokattavuuden, liitettävyyden ja edullisen hinnan takia.

Merkittävä puute Arduino-järjestelmissä automaation kannalta on tulojen ja lähtöjen rajoittuminen 0–5 V standardin puitteisiin. Ongelma voidaan korjata elektroniikan muuntokytkennöillä, mutta tämä ei ole se mieluisin eikä helpoin tapa tehdä asioita. 0–5 V standardia tukevia antureita ja ohjausyksiköitä on toki ole-

massa runsaasti, mutta ammattikäyttöön tarkoitetut laitteet tukevat monesti joko 0–10 V, 0–20 mA tai 4–20 mA standardia.

6.5 Arduinon liitettävyyys automaatiojärjestelmiin

Tyypillisesti Arduino-laitteisiin perustuvat rakennusautomaatiojärjestelmät toteutetaan Ethernet, Zigbee, Bluetooth tai näiden yhdistelmien avulla. Monet elektroniikan valmistajat ovat kehittäneet yhdistämiseen soveltuvia liitäntäyksiköjä (Shield) ja useat sovelluskehittäjät ovat itsenäisesti toteuttaneet avoimen lähdekoodin kirjastoja. Arduino-laitteet pystytään myös yhdistämään monenlaisiin automaatiojärjestelmiin.

RS 485 Shieldin avulla pystytään liittämään jopa 32 laitetta samanaikaisesti väylään ja samalla kaapelilla tiedonsiirto on mahdollista jopa yli kilometrin etäisyydelle. RS 485 antaa tuen kaksisuuntaiselle half duplex -liikenteelle ja mahdollistaa 10 Mbit/s tiedonsiirtonopeuden. liitäntäyksikön ja soveltuvan kirjaston avulla Arduino-piirikortti voidaan liittää esimerkiksi Modbus RTU-väylään. [11; 27.]

CAN Shield mahdollistaa Arduino-laitteiden liittämisen mm. henkilöautoissa ja teollisuudessa käytettyyn CAN-väylään. OBD-2-kaapelin avulla piirikortti voidaan ohjelmoida esimerkiksi toimimaan henkilöauton tiedonkeräys tai diagnostiikkalaitteena. CAN Shield tukee CAN V2.0B väylässä tiedonsiirtoa 1Mbit/s asti. [3.]

Ethernet Shieldin ja käyttäjien luoman Modbus-kirjaston avulla Arduino-alusta pystytään liittämään Modbus TCP-väylään. Myös Siemens S7 Ethernet -laitteiden ja Arduinon välisen liikenteen hallitsemiseen on olemassa oma kirjastonsa. Settimino.h nimisen kirjaston tukemana Arduino-kortit saadaan yhdistettyä Siemens S7 ja Logo! CPU -yksiköihin. Tällä hetkellä Settiminoa on testattu vain Uno R3 ja Mega R3 -korttien kanssa, mutta kirjaston pitäisi toimia myös muiden Arduino-piirikorttien kanssa. Settimino-projektin kotisivuilta löytyvät selkeät yksityiskohtaiset ohjeet eri laitteiden yhdistämiseen. [14.]

6.6 Arduinon tiedonsiirto lisälaitteiden kanssa

Arduinon pääasialliset tiedonsiirtotavat antureiden sekä muiden lisälaitteiden kanssa ovat sarjaliitettä (UART), I²C ja SPI-yhteydet.

Sarjaliitettä on yksi tärkeimmistä tiedonsiirtotavoista. Sarjamuotoiseen liikenteeseen perustuu useita eri standardeja ja tiedonsiirtotapoja. Arduino käyttää sarjaliikennettä joidenkin antureiden sekä tietokoneen välisiin yhteyksiin. Yleisimpiä UART-protokollaa käyttäviä antureita ovat RFID ja GPS-tekniikoihin perustuvat laitteet. Tiedonsiirto Arduinon ja tietokoneen välillä tapahtuu yleensä USB-väylän avulla. UART-protokollaan pohjautuvat anturit ja lisälaitteet keskustelvat Arduino-laitteiden kanssa TX/RX -liitäntöjen avulla. [12, s. 110.]

I²C on sulautettujen järjestelmien tietoliikenteeseen tarkoitettu standardi. I²C toimii master/slave periaatteella, jossa piirikortti toimii isäntänä ja anturi tai muu lisälaite orjana. I²C pystyy kaksisuuntaiseen half duplex -tiedonsiirtoon. Half duplex ei pysty siirtämään tietoa molempiin suuntiin samanaikaisesti ja sitä käytetäänkin yleensä vähäistä tietomäärää siirrettäessä. [12, s. 114.]

I²C-yhteyttä käytetään mikroprosessorien välisessä tiedonsiirrossa, mutta parhaiten se sopii kuitenkin digitaalisten antureiden ja piirikortin väliseen kommunikaatioon. I²C-protokollan isäntälaitteet tunnistavat orjalaitteet erityisen uniikin osoitteen avulla. Kaupallisissa I²C-orjalaitteissa tämä osoite on sisäisesti ohjelmoituna, mutta esimerkiksi kahden mikroprosessorin välisessä kommunikaatiossa tämä osoite täytyy määritellä erikseen. I²C-tiedonsiirto vaatii toimiakseen kaksi yhteyttä:

- SDA (Serial Data Line) – tiedonsiirtoa varten
- SCL (Serial Clock Line) – kellopulssi tiedonsiirron tahdistusta varten [12, s. 114.]

SPI on standardoitu ja suosittu protokolla laitteidenvälisessä kommunikaatiossa. SPI toimii I²C:n tapaan master/slave periaatteella. SPI kykenee kaksisuuntaiseen full duplex -yhteyteen. Kehittyneemmän kommunikaation ja nopeamman tiedonsiirron ansiosta tavallisia antureita monimutkaisemmat laitteet kytketään Arduinoon SPI:n avulla I²C-väylän sijasta. Yleisimpiä SPI-yhteyttä käyttäviä laitteita ovat WiFi ja Ethernet Shieldit. SPI kytketään Arduinon ICSP-liitäntään ja käyttöjännitteen lisäksi yhteys tarvitsee toimiakseen neljä kytkentää:

- MISO (Master In Slave Out) – tiedonsiirtoyhteys orjan lähettämää dataa varten
- MOSI (Master Out Slave In) – tiedonsiirtoyhteys isännän lähettämää dataa varten
- SCK (Serial Clock) – kellopulssi tiedonsiirron synkronointia varten
- SS (Slave Select) – laitekohtainen yhteys orjalaitteen ohjaamiseen. [12, s. 120–121.]

7 Työssä käytettävät anturit ja moduulit

7.1 Hiukkasanturi GP2Y1010AU0F

Sharp GP2Y1010AU0F -hiukkasanturi (kuva 7) on tarkoitettu esimerkiksi pölyn ja tupakansavun havaitsemiseen. Anturi kykenee mittaamaan hiukkasten määrää pienhiukkasista hengitettävien hiukkasten kokoluokkaan saakka. Anturi soveltuu erityisesti käytettäväksi ilmanlaadun tarkkailemiseen ja sen toimintaa sovelletaan ilmanpuhdistajissa, ilmastointilaitteissa ja muissa ilmanlaatua mittaavissa laitteissa. [18.]



Kuva 7. Sharp GP2Y1010AU0F -hiukkasanturi

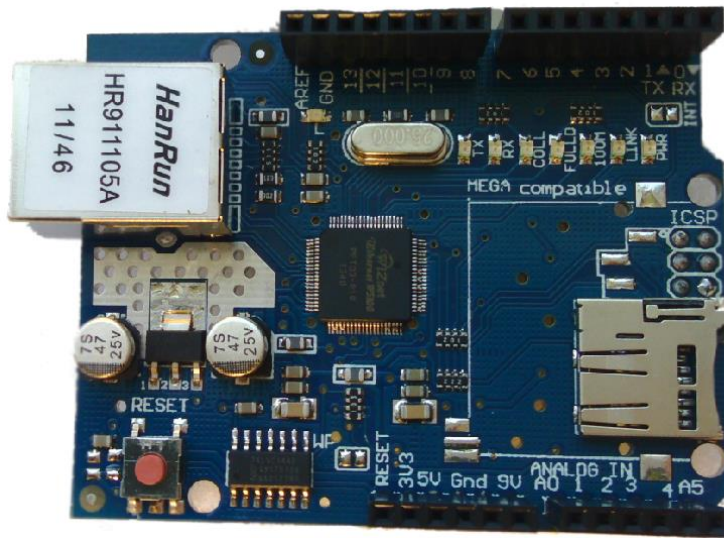
Hiukkasmittaus perustuu optiseen sirontamenetelmään, jossa hiukkasten tiheys arvioidaan infrapunasäteilyn määrän perusteella. Anturi koostuu infrapunasäteilyä lähettävästä IR-ledistä, vastaanottavasta IR-valotransistorista, ohjainyksiköstä ja signaalin vahvistajista. Anturin kotelossa on keskellä reikä, jonka läpi mitattava ilmavirta pääsee kulkemaan. Reiän eteen suositellaan asennettavaksi karkeaa suodatinkangasta, etteivät esimerkiksi suuret pölyhiukkaset pääse tukkimaan kotelon sisustaa. Tällä tavoin myös laitteen mittaamaa hiukkaskokoa pystytään rajoittamaan. [18.]

Hiukkasanturin ominaisuuksia:

- hiukkaskoko säädettävissä suodattimien avulla
- tarkkuus $0,1 \text{ mg/m}^3$
- käyttöjännite $-0,3...+7 \text{ V}$
- virrankulutus $<20 \text{ mA}$
- käyttölämpötila $-10...+65 \text{ °C}$ [18.]

7.2 Hanrun Ethernet Shield

Ethernet Shield (kuva 8) on tarkoitettu Arduino-laitteiden liittämiseksi Ethernet-verkkoon. Se perustuu Wiznet W5100-piiriin ja sisältää 16kt puskurimuistia. Kortti liitetään Arduinon ICSP-porttiin.



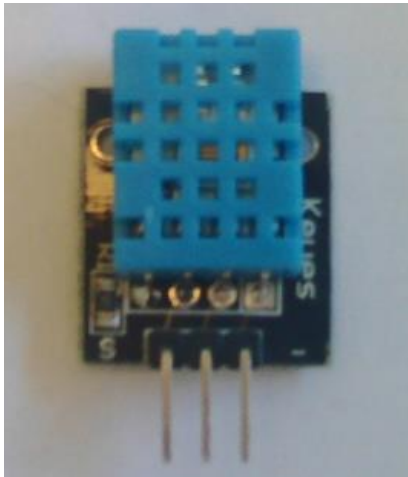
Kuva 8. Ethernet Shield

Ethernet Shieldin ominaisuuksia:

- tuki seuraaville TCP/IP protokollille : TCP, UDP, ICMP, IPv4 ARP, IGMP, PPPoE, Ethernet
- tiedonsiirtonopeus 10/100Mbps
- micro SD –kortinlukija [22.]

7.3 Kosteus- ja lämpötila-anturi DHT11

DHT11 (kuva 9) on Aosongin valmistama lämpötilan ja suhteellisen kosteuden mittaukseen tarkoitettu digitaalinen anturi. Sen käyttökohteita ovat ilmastointilaitteet, ilmankosteuttajat, sääasemat, ja muut mittalaitteet joiden tehtävänä on mitata ilmankosteutta sisälämpötiloissa. DHT11 sisältää 8-bittisen mikrokontrollerin, NTC-lämpötila-anturin ja kapasitiivisen kosteusanturin. [5.]



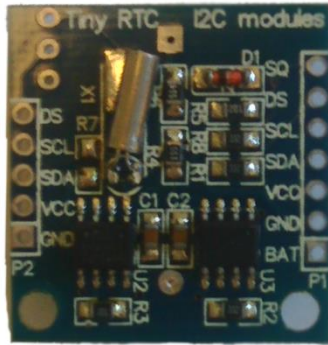
Kuva 9. DHT11

DHT11:n ominaisuuksia:

- digitaalinen binäärinen ulostulo
- suhteellisen kosteuden mittaust 20...95 % välillä 1 % resoluutiolla ja ± 5 % tarkkuudella
- lämpötilan mittaust 0...50°C välillä 1°C resoluutiolla ja $\pm 2^\circ\text{C}$ tarkkuudella
- käyttöjännite +3...+5VDC
- virrankulutus 60 μA levossa ja 0,3mA mittauksen aikana [5.]

7.4 Reaaliaikakellomoduuli DS1307

DS1307 Reaaliaikakello (kuva 10) on paristovarmistettu kalenteriominaisuuksilla varustettu kellomoduuli. Yleisimmät käyttökohteet ovat sulautettujen järjestelmien aikatoiminnot.



Kuva 10. DS1307

Kellon ominaisuuksia:

- kommunikointi I²C-väylän kautta
- virransyötön tunnistus ja automaattinen varavirtaan siirtyminen
- virrankulutus <500nA
- käyttölämpötila -40...+85°C
- käyttöjännite +4,5...+5,5VDC
- ajan näyttö sekunnin tarkkuudella [19.]

8 Järjestelmän rakentaminen ja ohjelmointi

Tässä luvussa käydään läpi laitteen piirikortille liitettävien antureiden ja moduulien ominaisuuksia, liittämistä, kalibrointia, sekä ohjelmointia. Lisäksi luvussa käsitellään ilmastoinnin säädön toteuttamista, laitteen kotelointia ja yhdistämistä Xively-palveluun.

8.1 Reaaliaikakellon liittäminen ja ajastus

DS1307-reaaliaikakellomoduuli liitettiin Arduino-piirikortille I²C-väylään SDA ja SCL-liitäntöihin. Kellomoduulin piirilevyllä oleva akkuparisto pitää huolen aikatiedon säilymisestä, mutta alussa aikatieto syötetään manuaalisesti ohjelmointiympäristön avulla. Kello synkronoitiin valmiin esimerkkiohjelman avulla.

Valmiissa ohjelmassa ilmeni ensin hieman outo ominaisuus. Kello näytti 00–09:n väliset tunnit ja minuutit ilman edessä olevaa nollaa ja täytti jäljellä olevan muistin sekalaisilla numeroilla. Tämä korjattiin lisäämällä ohjelmaan muutama määrittäysrivi.

8.2 Lämpötilan ja suhteellisen kosteuden mittaaminen

DHT11:n mikrokontrollerin lähettämä sanoma koostuu 40 bitin pituisesta bittijonosta ja sanoman rakenne koostuu kuvan 11 mukaisesti.

0001 1000 + 0000 0000 + 0010 0110 + 0000 0000 = 0011 1110
 Kosteus% Kosteus% desimaalit Lämpötila Lämpötila desimaalit Pariteettitavu

Kuva 11. DHT11:n digitaalisen sanoman rakenne

Lämpötila ja kosteustiedoille on molemmille varattu 16-bittiset sanomat, jossa ensimmäiset 8 bittiä kertovat kokonaisluvut ja seuraavat 8 bittiä desimaaliluvut. DHT11 ei tosin epätarkkuutensa vuoksi tue desimaalien esitystä, joten nämä bitit ovat todellisuudessa aina nollia. Pariteettitavu ilmoittaa mitatun tiedon oikeellisuuden. Jos pariteettitavun summa ei vastaa lämpötila- ja kosteustietojen yhteenlaskettua arvoa, niin mittaustiedot ovat virheelliset ja anturilta odotetaan uutta mittaustietoa.

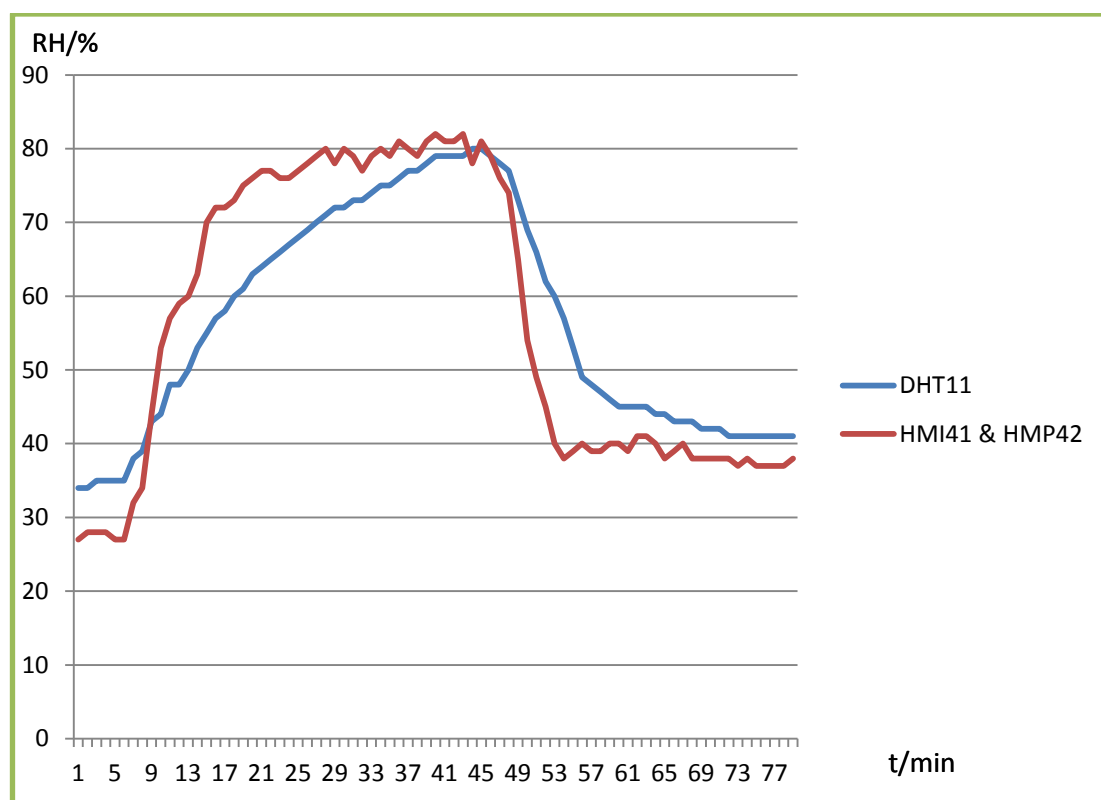
Kyseisessä tapauksessa (kuva 11) mitatut arvot olisivat:

- kosteus: 0001 1000, 0000 0000 = 18H = 24 %
- lämpötila: 0010 0110, 0000 0000 = 26H = 38°C , jolloin pariteetin arvoksi pitäisi tulla 24D + 38D = 62D = 3EH
- pariteetti: 0011 1110 = 3EH = 62D, joten esimerkkitapauksessa saatiin paikkansapitävä mittaustieto. [5.]

DHT11-anturit sijoitettiin poistoilmakanavaan ja eteiseen. Toinen anturi päädyttiin sijoittamaan eteiseen, koska perimmäisenä ajatuksena oli mitata huoneil-

man kosteutta ja verrata sitä poistoilman kosteuteen. Anturit ovat tehdaskalibroituja, mutta mittaustarkkuuden optimoimisen vuoksi kosteusmittaukselle oli tarpeellista suorittaa ohjelmallinen kalibrointi. Kalibrointia varten luotiin luonnolliset olosuhteet.

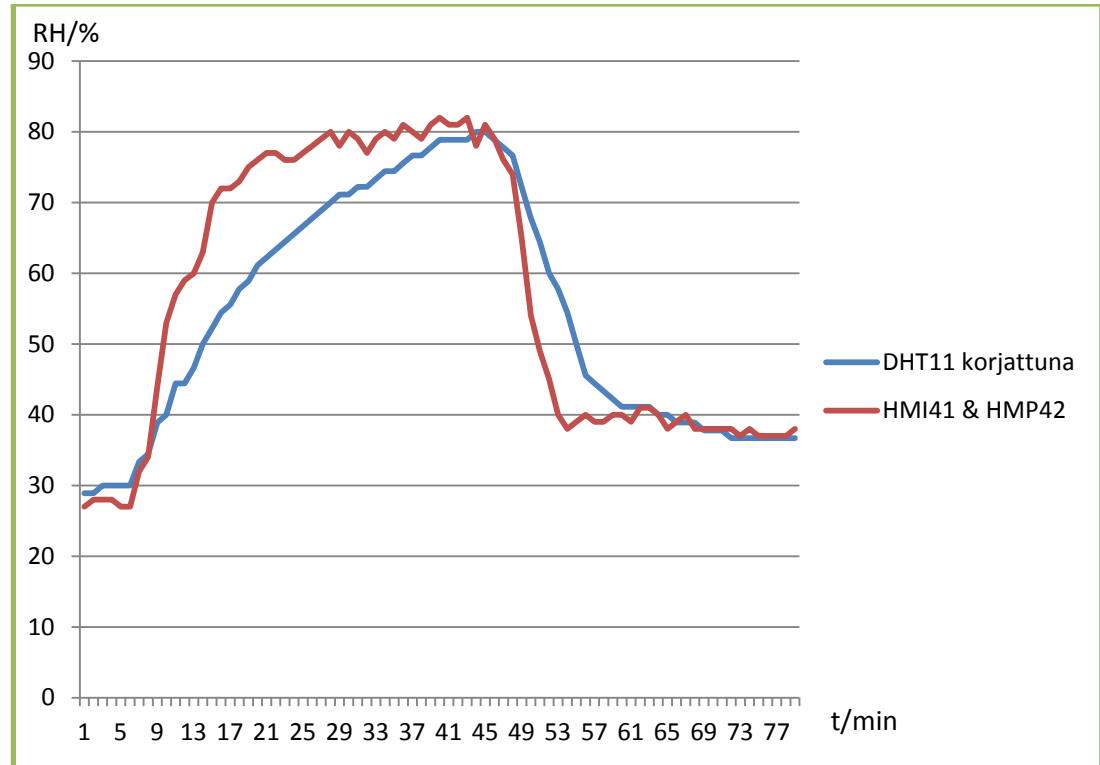
Kalibroinnin apuvälineenä käytettiin Vaisala HUMICAP® HMI41-mittalaitetta HMP42-mittapäällä varustettuna. Mittalaite kalibroitiin aiemmin samana päivänä litiumkloridi (LiCl), 11 % RH ja kaliumsulfaatti (K_2SO_4), 97 % RH liuoksia käyttäen Vaisala HMK15-kosteuskalibraattorin avulla. Ennen DHT11-anturin kalibrointia sekä HMI41-laitteen, että DHT11-anturin annettiin asettua noin tunnin verran mittauksen suoritustilassa.



Kuva 11. Suhteellisen kosteuden mittaukset

DHT11 mittaa suhteellista kosteutta 20–95 %:n väliltä ja kalibrointi täytyi suorittaa huone/poistoilman tyypillisissä kosteusolosuhteissa. Mittaus suoritettiin kylpyhuoneessa ja kosteutta lisättiin ilmaan suihkun avulla. Suhteellisen kosteuden arvo vaihteli mittauksessa 27–81 %:n välillä. DHT11-anturi reagoi hitaammin muutoksiin kuin HMP42, mutta pidemmällä aikavälillä mittaustulokset kuitenkin

tasaantuivat samoihin lukemiin. DHT11 näytti kuivemmissä olosuhteissa liian korkeaa RH-arvoa (kuva 11), joten mittausarvolle määritettiin ohjelmallinen korjaus.



Kuva 12. Korjatut mittausarvot

Korjaus suoritettiin seuraavien kaavojen avulla:

Laskettiin ensin korjauskerroin k_r

$$k_r = \frac{RH_{max}}{RH_{max} - 8} = \frac{80}{72} = 1,1111 \quad (2)$$

missä RH_{max} = korkein mitattu suhteellinen kosteus

Korjattu kosteusarvo RH_{real} saatiin siis kaavalla:

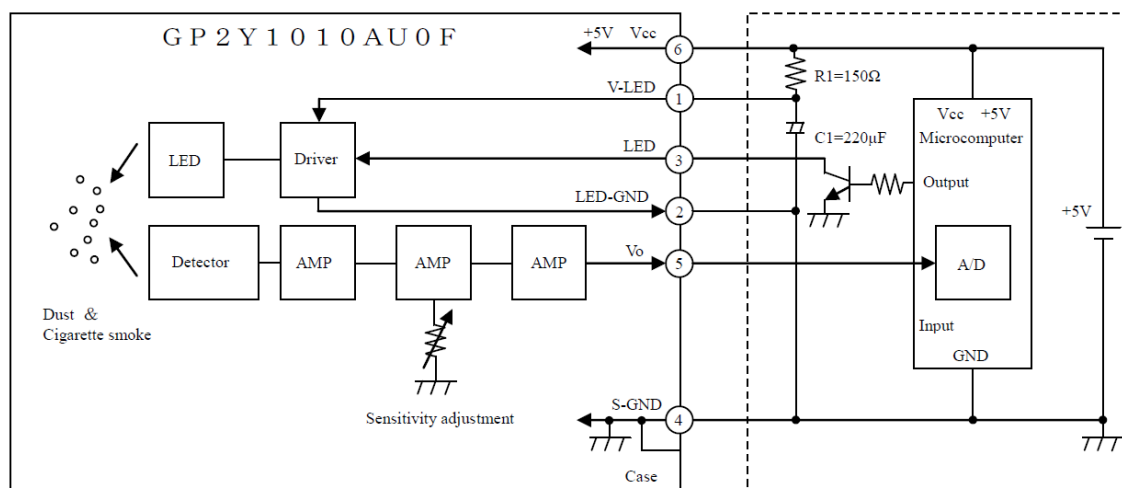
$$RH_{real} = k_r * (RH_{meas} - 8) \quad (3)$$

missä RH_{meas} = mitattu kosteusarvo

Edellä mainitun korjauskertoimen käyttäminen tuottaa kuvassa 12 näkyvät tulokset. Periaatteessa kosteusmittauksen muutosta voisi nopeuttaa lisäämällä ohjelmaan derivoivan tekijän. Derivointi aiheuttaisi joissain tilanteissa mittauksiin ylilyöntejä, joten ko. prosessiin se on tarpeeton. Myös korjauskertoimen käyttö aiheuttaa virhettä yli 80 %:n suhteellisissa kosteuspitoisuuksissa. Sisäilman suhteellisen kosteuden tavoitearvot ovat vuodenaajoista riippuen 25–60 %, joten DHT11:n mittaustarkkuuden optimointi juuri tälle tarkasteluvälille on perusteltua. Kostean huoneilman vuoksi mittauksessa ei valitettavasti saavutettu alle 25 % RH-olosuhteita.

8.3 Hiukkaspitoisuuden mittaaminen

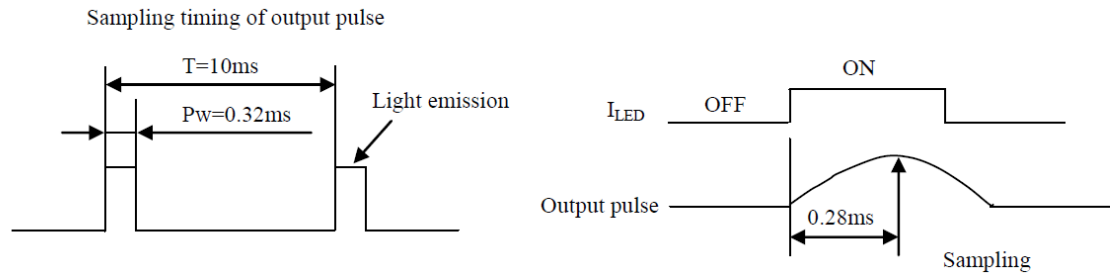
GP2Y1010AU0F vaatii pulssiajon (pulse drive) takia toimiakseen ulkopuolisen vastuksen ja kondensaattorin kytkennän (kuva 13). Anturin V-Led napaan kytkettiin jännite 150Ω vastuksen kautta. Vastuksen kanssa sarjaan kytkettiin 220μF elektrolyyttikondensaattori. Anturin signaalilähtö kytkettiin piirikortin analogiatuloon.



Kuva 13. Hiukkasanturin kytkentä [18]

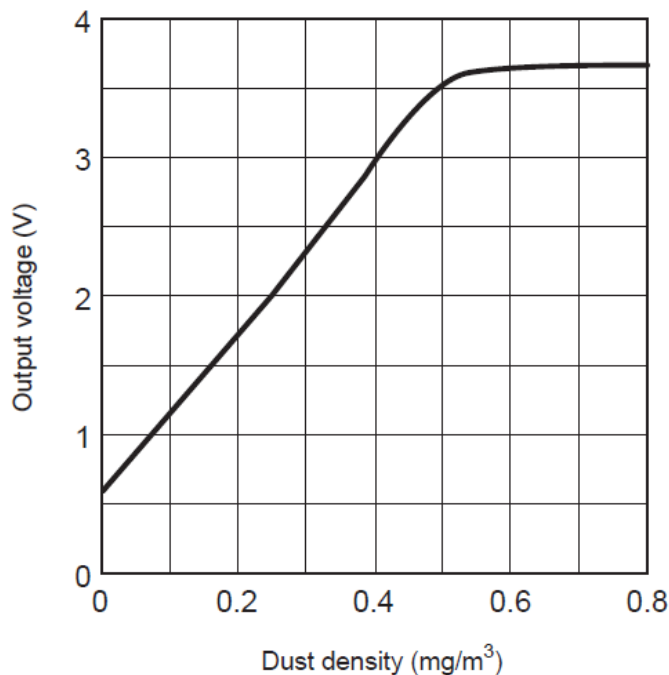
Anturia asennettaessa täytyi ottaa huomioon tiettyjä olosuhteita. Mittauksiin voi syntyä häiriöitä ulkoisesta infrapunasäteilystä ja induktiivisista laitteista. Myös ilmanvaihtokoneen puhaltimien moottoreista olisi saattanut aiheutua induktiivis-

ta häiriötä, joten anturi sijoitettiin tarpeeksi kauas niistä. Lopullinen sijoituspaikka anturille löytyi eteisestä.



Kuva 14. Anturin näytteenoton suositeltu ajastus [18]

Anturin näytteenotto suositellaan suoritettavaksi kymmenen millisekunnin jaksoissa siten, että signaalin alussa IR-lediä pidetään päällä 0,32 ms. 0,28 ms:n kohdalla kaapataan sen hetkinen mittaustulos ja lähetetään se mikrokontrollerille. Lopuksi pidetään 9,68 ms:n pituinen tauko, jonka jälkeen aloitetaan uusi näytteenotto (kuva 14).



Kuva 15. Ulostulojännitteen riippuvuus hiukkasten määrästä [17]

Hiukkaspitoisuutta mitattiin useita päiviä eri olosuhteissa ja jännitteen arvoiksi saatiin tuloksia vaihteluväliltä 0,6V–3,7 V. Saadut tulokset mahtuvat juuri kuvassa 15. esitettyjen referenssiarvojen puitteisiin. 0,6 V jännitearvo saavutettiin

pölyttömissä olosuhteissa ja 3,7 V arvo suihkutettaessa ilmaan runsaasti aerosolia. Koska jännitteen ja hiukkaspitoisuuden suhde on lineaarinen välillä 0,6 V ja 3,4 V, niin tulokset voidaan muuttaa $\mu\text{g}/\text{m}^3$ -muotoon seuraavalla tavalla:

Kerroin k_a saadaan kaavasta 4:

$$k_a = \frac{(470\mu\text{g}/\text{m}^3)}{\left(\frac{3,4\text{V}}{5\text{V}} * 1023\right) - \left(\frac{0,6\text{V}}{5\text{V}} * 1023\right)} = \frac{(470\mu\text{g}/\text{m}^3)}{(695,64 - 122,76)} = 0,820416 \quad (4)$$

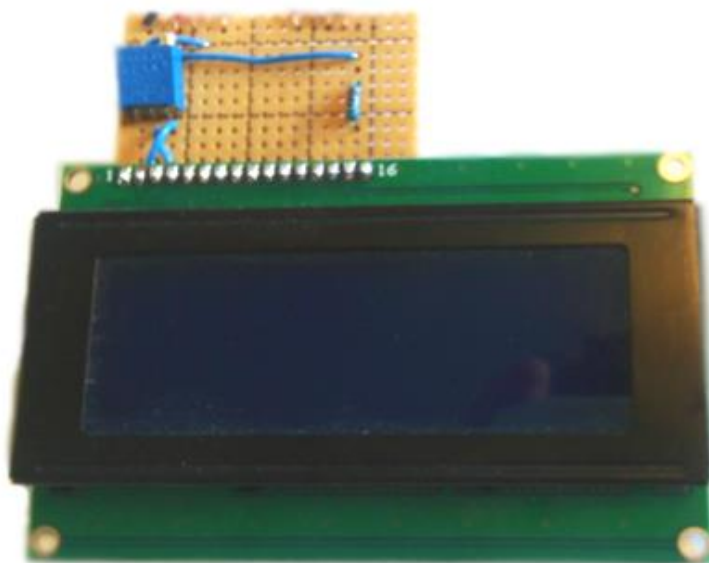
Ohjelmassa hiukkasmittauksesta saatu tulos muunnetaan näyttämään $\mu\text{g}/\text{m}^3$ -arvoa $D_{\mu\text{g}}$ yhtälöllä 5:

$$D_{\mu\text{g}} = (D_{\text{val}} - 122,76) * k_a \quad (5)$$

missä D_{val} = analogiatulon jännitettä vastaava lukuarvo

8.4 LCD-näytön liittäminen

Hitachi HD 44780-yhteensopivat näytöt ovat edullisia, suosittuja ja sopivat hyvin monenlaisiin projekteihin. Projektiin valittiin 80-merkkinen nelirivinen malli, joka riittää hyvin mittausrvojen ilmoittamiseen. Näytön mukana ei tullut liitäntäyksikköä, joten näytön liitäntäpinnit ja tarvittavat vastukset juotettiin piirilevylle.



Kuva 16. Hitachi HD44780-yhteensopiva LCD-näyttö

Kuvassa 16 oleva $10\text{k}\Omega$ trimmeripotentiometri on tarkoitettu näytön kontrastin säätämiseen ja 220Ω metallikalvovastus taustavalon etuvastukseksi. Näyttö liitettiin Arduino-kortin digitaaliulostuloihin. Piirikortin ja näytön välinen tiedonsiirto tapahtuu rinnakkaismuotoisina neljän bitin sanomina. HD44780-yhteensopiville näytöille on olemassa I²C-sarjaliikennettä käyttäviä rinnakkais/sarja muuntimia, mutta tässä työssä sellaiselle ei ollut tarvetta. Näytön määrittelyjä varten ohjelmaan sisällytettiin Liquidcrystal.h -kirjasto.

8.5 Liittyminen Ethernet-verkkoon

Ethernet Shield pystyy toimimaan joko asiakas (Client) tai palvelin (Server) moodissa. Palvelintilassa kortti pystyy esimerkiksi isännöimään nettisivuja ja asiakastilassa voidaan noutaa tai lähettää tietoja palvelimille. Tätä ohjelmaa varten Shield ohjelmoitiin toimimaan asiakkaana.

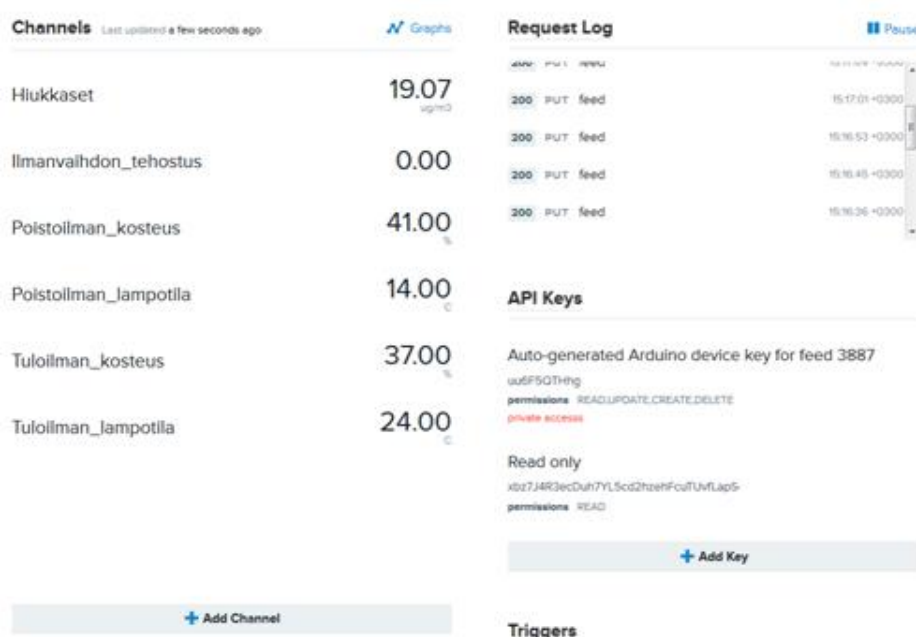
Tämän mallin verkkokortti ei sisällä "rautakoodattua" MAC-osoitetta, joten se täytyy määritellä ohjelmallisesti ennen kuin yritetään saada yhteyttä verkkoon.

Ethernet.h -kirjasto sisältää automaattiset määrittelyt DHCP-toimintoa varten, joten muita ohjelmallisia määrittelyjä ei ollut tarpeellista suorittaa internet-yhteyttä varten. Liitäntäyksikkö toimii SPI-yhteydellä ja se liitettiin Arduino-alustan ICSP-liitäntään.

8.6 Kommunikointi Xively-palvelimen kanssa

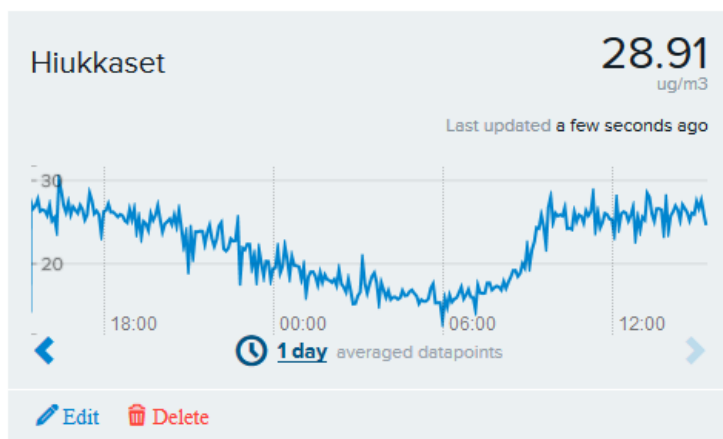
Liittyminen Xively-pilvipalveluun oli suhteellisen helposti toteutettavissa. Verkkosivuille tehtiin tunnukset, jonka jälkeen pystyttiin suoraan määrittelemään palveluun liitettäviä laitteita. Arduinon ja Xivelyn välinen kommunikointi on http-protokollan mukaista, mutta salasanalla suojattua.

Ohjelmaan sisällytettiin Xively.h -kirjasto, joka määrittelee yhteyden laitteen ja palvelimen välillä. Ohjelmaan tehtiin myös yksinkertaisia määrittelyjä yhteyttä varten. Xively luo yhdistettävää laitetta varten Feed ID, sekä API key tunnukset, joista Feed ID toimii syötteen käyttäjätunnuksena ja API key salasanana. Käyttäjä voi myös itse luoda palvelussa eritasoisia API-avaimia. Käyttäjä voi esimerkiksi määritellä API-avaimen voimassaoloajan, luku- ja kirjoitusoikeudet, sekä sallitun Ip-osoitteen. Tämä parantaa palvelun turvallisuutta.



Kuva 17. Xively-palvelun käyttäjänäkymä

Xivelyn perusnäky (kuva 17) on selkeä ja riisuttu. Käyttäjä näkee sivulta tunnukseensa liitetyt kanavat, yhteydet, triggerit ja muut tarpeelliset tiedot. Palvelu myös piirtää kerätyistä tiedoista pyydettyä graafista näkymää (kuva 18).



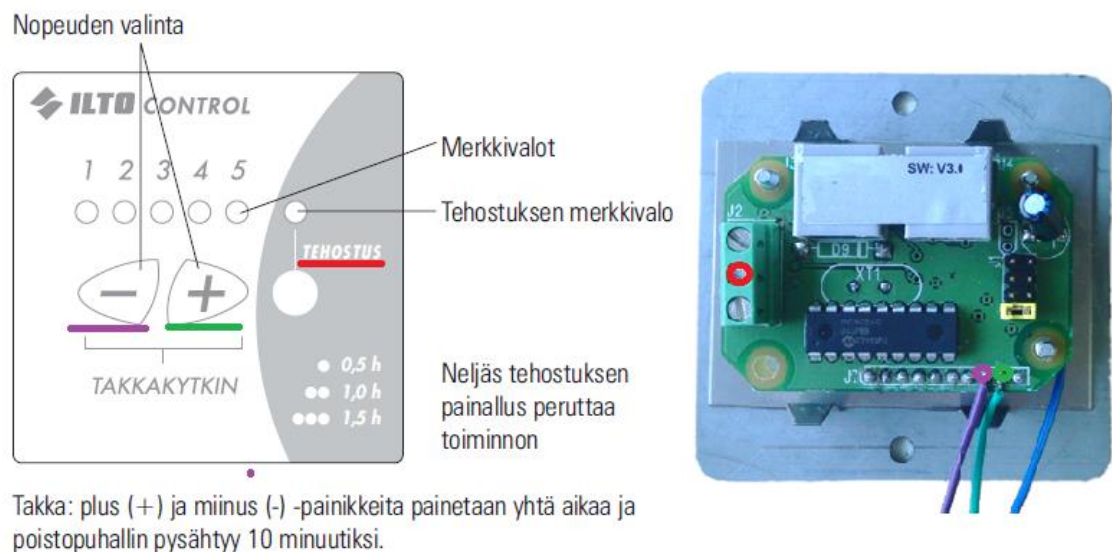
Kuva 18. Esimerkki hiukkasmittauksen graafisesta näkymästä (Xively)

8.7 Ilmastoinnin säätö

Alkuperäisessä suunnitelmassa tarkoituksena oli ohjata poistopuhaltimen pyörimisnopeutta koneen sisäiseltä piirikortilta, jossa on valmis liitäntä 0–10 V jänniteviestille. Tällä tavoin ohjaus olisi onnistunut Arduinon PWM-lähdön ulostulojännitteen muuntokytkennän avulla. Laitetta tarkemmin tarkastellessa huomattiin sähköliitännöiden piirikortin olevan ruuvattu tiukasti koteloon monesta kohdasta. Piirikorttiin käsiksi pääseminen olisi vaatinut lämmön talteenottoyksikön irrottamista ja kotelon purkamista. Koska sen kaltainen ratkaisu ei kohteessa tullut kyseeseen, päädyttiin etsimään vaihtoehtoisia tapoja koneen ohjaamiseen.

Ohjaus toteutettiin lopulta eteisessä olevan säätöpaneelin mikrokontrollerin liitäntänapoihin tehtyjen kytkentöjen avulla. PIC 16C54C-mikrokontrollerin käyttöjännite on 5 VDC, ja toimintoja pystytään siten ohjaamaan Arduinon digitaalisen lähdön avulla suoraan. Tämä vaati ainoastaan Arduinon ja säätöpaneelin maapotentiaalien yhdistämistä.

Kuvassa 19. näkyvät työtä varten juotetut johtimet sekä värikoodeilla selitetyt toiminnot. Vihreään/violettiin johtimeen annettu signaali tehostaa/laskee puhaltimen nopeutta yhdellä pykälällä ja punaiseen liittimeen johdettu signaali käynnistää kahden pykälän tehostuksen puolen tunnin sykleissä. Sininen johdin on kytketty maapotentiaaliin. Tarvittavat lisätiedot mikrokontrollerista saatiin PIC16C54C–datalehddestä [24]. Kytkenät selvitettiin yleismittarin resistanssimittauksen avulla.



Kuva 19. Ilmastointilaitteen käyttöpaneelin toiminnot ja liitännät

Ilmastoinnin säätö toteutettiin käyttäen kaksiporista säätöä. Ilmankosteus vaihtelee runsaasti vuodenajasta riippuen, jonka vuoksi olisi vaikeaa määrittää jotakin tiettyä kosteuden raja-arvoa, joka käynnistäisi ilmastoinnin tehostuksen. Ongelma ratkaistiin mittaamalla kosteus sisäilmasta ja vertaamalla sitä poistoilman kosteuteen. Ohjelmaan asetettiin lopulta sopiva kosteuseron raja-arvo, joka käynnistää ilmanvaihdon tehostuksen.



Kuva 20. Poistoilman kosteus ilmanvaihdon ollessa normaaliasennossa (Xively)

Poistoilman kosteus mitattiin ilmastointilaitteen poistoilmakanavasta. Poistoilman suhteellinen kosteus nousi kylpemisen aikana normaalikäytön nopeuksilla jopa 87 %:in ja laski normaaliin kosteuteen vasta monen tunnin päästä (kuva 20). Kylpyhuoneessa ilmanvaihdon puutteellisuus huomattiin kosteuden tiivistymisestä seinien ja ikkunoiden pinnoille. Sisäilman kosteus eteisestä mitattuna vaihteli tarkasteluvälillä 34–38 %:n rajoissa.



Kuva 21. Poistoilman kosteus ilmanvaihdon ollessa tehostettuna (Xively)

Ilmanvaihdon ohjelmoidun tehostuksen aikana poistoilman suhteellinen kosteus nousi korkeintaan 54 %:n ja palasi takaisin normaaliin alle tunnissa. Sisäilman kosteus vaihteli tarkasteluvälillä 34–36 %:n rajoissa (kuva 21).

Ilmanvaihdon tehostus ohjelmoitiin käynnistymään vähintään 7 %:n suhteellisen kosteuden erosta. Tehostus jatkui niin kauan, kunnes kosteusero alitti 7 %:n arvon. Ilmanvaihdon ollessa tehostettuna kosteus ei tiivistynyt pinnoille.

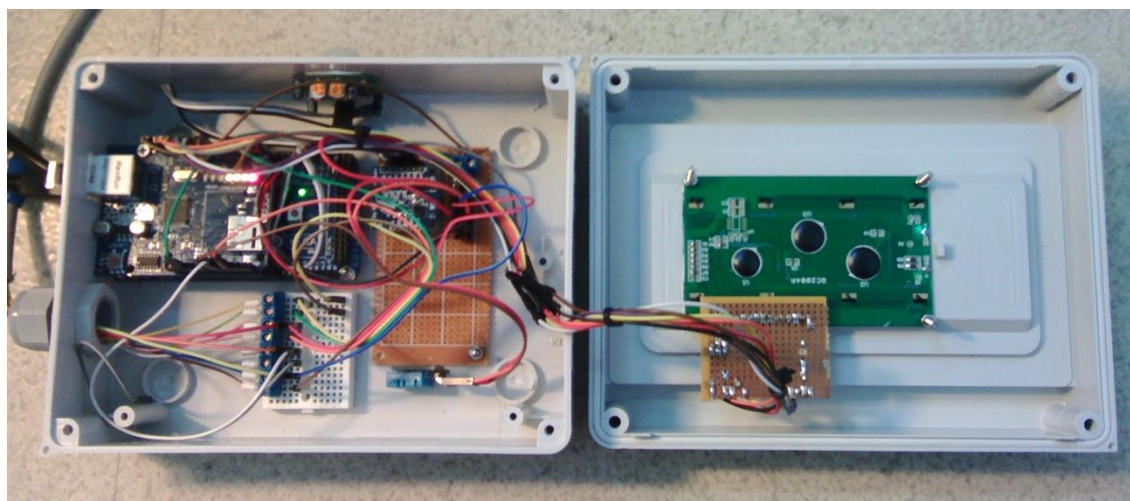
Tehostustoiminto on kiinteästi ohjelmoitu käyttöpaneelin elektroniikkaan ja toimituksessaan se lisää puhaltimien nopeutta kahdella pykälällä. Tehostus toimii puolen tunnin sykleissä siten, että jokainen signaalin nouseva reuna lisää puoli tuntia tehostettua aikaa puoleentoista tuntiin saakka. Neljäs nouseva signaalin reuna nollaa tehostustoiminnon.

Koska mittaustietoa halutaan jatkuvasti, ajastustoiminto ei saa vaikuttaa ohjelman muuhun toimintaan. Tämä aiheuttikin ohjelman rakenteeseen suuren muutoksen. Ohjelma ajastettiin siten, että pääsilmukan toiminnot suoritetaan 60 sekunnissa. Käyttöön otettiin apumuuttuja, jonka arvoa kasvatetaan jokaisella yli 7

%:n kosteuseron saavuttavalla silmukan kierroksella. Kun apumuuttuja saa arvon 1, niin tehostustoiminto aloitetaan. Jos kosteusero laskee alle 7 %:n tai apumuuttuja saa arvon 31, niin apumuuttujan arvo nollataan. Täten varmistetaan, että tehostus toimii puolen tunnin sykleissä, eikä ohjelma pyöritä tehostuksen silmukkaa jatkuvasti ympäri. Ilmastointilaitteen käyttöpaneelin muita ohjauksia on myös mahdollista käyttää Arduino-kortin digitaalilähtöjen avulla.

8.8 Kotelointi

Laitteiston kotelointia varten hankittiin käyttötarkoitukseen sopiva muovinen laitekotelo. Koteloon tehtiin läpiviennit liitäntöjä varten. Kaapelointiin käytettiin suojattua LIYCY 8X0 -tiedonsiirtokaapelia, joka vietiin holkkitiivisteiden läpi koteloon. Laitteiston sisäisissä johdotuksissa käytettiin dupont-johtimia (kuva 22). Laitteiston sisäisten kytkentöjen piirikaavio esitetään liitteessä.



Kuva 22. Laitteen kotelointi avattuna

Laitteesta tuli ulkoasultaan siisti. Laitekotelo (kuva 23) oli alkujaan IP65-luokiteltu, mutta piirikortin liitäntöjä varten tehdyt läpiviennit laskivat luokituksen arviolta IP30-luokkaan. Vesitiivis kotelo ei enää ole, mutta ilman työkaluja siihen ei edelleenkään pääse käsiksi. Laitetta on tarkoitus käyttää vain kuivissa sisätiloissa, joten suojaus on riittävä.



Kuva 23. Laitteen ulkoasu

9 Testaus

Järjestelmän toimintaa testattiin useita päiviä erilaisissa olosuhteissa. Lämpötilamittaukset pitivät paikkansa riittävän tarkasti ja kosteusmittaukset ovat tarkoituksenmukaisia käyttötapaan nähden. Kalibroinnilla saatiin parannettua kosteusmittauksen paikkansapitävyyttä ja mielestäni anturit soveltuvat tähän käyttötarkoitukseen hyvin.

Hiukkasmittaus näyttää suuntaa antavan arvion hiukkaspitoisuudesta. Mittausten todenmukaisuutta oli hankalaa todeta ilman asianmukaisia verrokkilaitteita. Anturi kuitenkin reagoi selvästi muutoksiin hiukkaspitoisuudessa. Myös mitatesa pitoisuutta ulkoilmasta arvot olivat samaa tasoa www.joensuu.fi -sivuston lähimmän hiukkasmittauspisteen vuorokausikeskiarvoihin nähden.

Laitteen yhteys verkkoon katkeili satunnaisilla aikaväleillä testatessa yhteyttä mobiililaajakaistaan liitettynä, mutta kiinteää laajakaistayhteyden ollessa saata-

villa tätä ongelmaa ei esiintynyt. Mysteeriksi jäi se, mihin yhteysongelma liittyi, mutta kokemusten perusteella lienee syytä epäillä kaistan huonoa kuuluvuutta alueella. Ohjelman toimintaan yhteyden satunnainen katkeaminen ei onneksi vaikuta, vaan mittaus sekä ohjaus toimivat normaalisti yhteyden ollessa poikki. Ainoa poisjäävä osuus on siis seuranta internetin välityksellä. Xively leimaa saapuvan datan saapumisajan mukaan, joten datan lähettäminen viivästetysti aikaleimoilla varustettuna ei ainakaan perusasetuksilla tule kyseeseen.

10 Laitteiston laajennusmahdollisuudet

Laitteistolle toteutettavissa olevia jatkokehitysmahdollisuuksia on useita. Yksi mahdollisuus laajentaa järjestelmän toimintaa olisi rakentaa ilmastointilaitteen sisään suodatinvahdin käyttämällä kahta barometristä paineanturia. Ilmanlaadun tarkkailuun saataisiin monipuolisuutta lisäämällä järjestelmään esimerkiksi hiilidioksidi- tai hiilimonoksidimittauksen.

Kohdekiinteistössä on saunan lämmitystä varten elektroninen käyttöpaneeli varustettuna saunatilan lämpötilamittauksella. Yksi mielenkiintoinen jatkokehitysmahdollisuus olisikin liittää Arduino tähän käyttöpaneeliin ja hallita saunan lämmitystä internetin välityksellä.

11 Kustannukset

Kustannukset haluttiin luonnollisesti pitää kohtuullisina ja suurin osa tarvikkeista tilattiin Ebay-verkkokaupasta. Hintaa prototyypin tarvikkeille muodostui taulukon 3 mukaisesti ja kokonaiskustannus jäi alle 90 euron.

Taulukko 3. Järjestelmän kustannukset

Tuote	Kpl	Hinta €
Arduino Mega 2560	1	20,87
Laitekotelo	1	12,99
Ethernet Shield W5100	1	7,80
DHT11	2	4,57
GP2Y1010AU0F	1	5,98
LCD 4x20	1	13,37
Muuntaja	1	7,99
DS1307	1	2,29
Johdot ja muut pientarvikkeet		14,00
Yhteensä		89,86

12 Pohdinta ja johtopäätökset

Arduino-laitteen rakentelu ja ohjelmointi olivat suhteellisen haastavaa, mutta palkitsevaa. Alan kirjallisuudesta ja internetistä löytyi valtavasti tietoa aiheesta, ja opinnäytteen aiheen rajaamisessa meinasikin tulla aluksi hieman ongelmia. En ollut käyttänyt Arduinoa ennen tämän työn aloittamista ja järjestelmän opettelussa menikin melkoisesti aikaa. Nyt laitteen valmistuttua on kuitenkin hienoa nähdä järjestelmä toiminnassa.

Olisin toivonut voivani ohjata ilmastointia suoraan ilmastointilaitteen piirikortilta jänniteviestin ja vaikka ohjelmallisen PID-säätöblokin avulla. Ilmastoinnin säätö kuitenkin onnistui hyvin myös seinäpaneelin mikroprosessorin digitaalituloja hyväksi käyttäen.

Jos tekisin työn nyt uudestaan, niin tilaisin lämpötila ja kosteusmittaukseen halvempien DHT11-mallien tilalle tarkemmat DHT22-anturit. DHT11-antureiden mittaamat lämpötila- ja kosteusalueet riittävät sisäilman lämpötilan ja kosteuden mittaamiseen hyvin, mutta suurempi tarkkuus ja laajempi mitta-alue olisivat toivottu lisä ja tekisivät mittaustiedoista merkityksellisempiä.

Laitteen toiminnan kannalta myös jotkin halvemmat Arduino-malliston piirikortit olisivat käyneet ominaisuuksiltaan aivan hyvin ja esimerkiksi UNO-mallin muisti olisi riittänyt ohjelman tallentamiseen. Ohjelman koosta ei kuitenkaan ollut tarkkaa ennakkokäsitystä ja tämän takia pitikin varautua rakentamiseen suuremman muistin omaavan Mega-piirikortin hankkimisella.

Erityisen positiivista työn tekemisessä oli uusiin laitteisiin, antureihin ja ohjelmointikieleen tutustuminen sekä oppiminen. Kokemuksena työn tekeminen oli tärkeä. Työn suorittamisessa pystyin hyödyntämään koulutusohjelmani oppeja soveltaen ja laaja-alaisesti.

Ammattikäyttöön ja haasteellisempiin prosesseihin Arduinon soveltuvuus on kyseenalaista, mutta harraste- ja kotikäyttöön se on mielestäni oiva järjestelmä. Yhteensopivia antureita ja laajennuskortteja on olemassa erittäin paljon ja niitä kehitetään lisää jatkuvasti. Alustan avoimuus on muokattavuuden, hallittavuuden ja verkottumisen kannalta mahtava asia, mutta toisaalta laitetuki jää pitkälti foorumeiden kautta toimivan vertaistuen varaan.

Arduino-laitteiden edullisuus ja käytettävyys tuovat laitteet lähes kaikkien saataville. Alhaisen käyttöjännitteensä vuoksi Arduinon kytkennät ovat sallittuja maallikoillekin. Parhaan potentiaalin Arduinoon pohjautuviin järjestelmiin näen langattomia antureita ja etäohjattuja toimintoja tarvitsevilla kohteilla. Tällaisia ratkaisuja tarvitaan usein mm. saneerauskohteiden automaatioissa.

Arduino sopii mainiosti ohjelmointia osaavalle, ja erilaisiin projekteihin löytyy paljon kirjallisuutta niin sähköisessä kuin paperisessa muodossakin. Arduino-laitteet ohjelmoidaan yleensä tekstipohjaisesti, ja tällä visuaalisten ohjelmointikielten aikakaudella se voi tuntua hieman vanhanaikaiselta. Mielestäni tekstipohjainen ohjelmointi kuitenkin onnistuu ilman suurempia ongelmia. Valmiit kirjastot mahdollistavat tietojenkäsittelyn hieman ylemmällä tasolla, eikä esimerkiksi kaikkia mahdollisia muistin ja porttien alustuksia tarvitse suorittaa alusta asti. Tunsin ennestään C-kielen perusteet ja vaikkapa AVR-ohjelmointiin verrattuna Arduinon kanssa toimiminen tuntui loogiselta ja tutulta.

Xivelyn ja Arduinon välinen kommunikointi toimii kokemuksien mukaan hyvin. Arduino Megan prosessointiteho ei riitä monimutkaisien salausten suorittamiseen, ja esimerkiksi siirrettävän liikenteen SSL-salausta varten tarvittaisiin ulkoisen laitteen apua. Http-liikenteenä ei kannata siirtää salaiseksi tarkoitettua tietoa, mutta ainakaan tässä sovelluksessa ei ollut tarkoituksena lähettää arkaluontoista dataa.

Arduino Mega ei tue moniajtoa, joten jatkuva porttien tarkkailu on monimutkaisemmissa ohjelmissa melko vaikeaa ja ohjelmoinnissa voi joutua siksi käyttämään erikoisia menetelmiä. Jos samanaikaisesti pystyisi ajamaan montaa loop-silmukkaa, niin ohjelmointi helpottuisi huomattavasti.

Suurin puute Arduino-laitteissa automaation kannalta on mielestäni laitteiden rajoittunut liitettävyyys. Jos piirikortti tulisi vaikka valmiiksi IP45-luokitellusti kotoituna ja varustettuna galvaanisesti erotetuilla 0–10 V standardia tukevilla I/O liitännöillä, niin voitaisiin puhua todella kilpailukykyisestä laitteistosta. Tosin Arduino-perheen laitteethan ovat tarkoitettu lähinnä prototyyppien rakentamiseen ja harrastustoimintaan.

Järjestelmän toiminta yltää sille asetettuihin tavoitteisiin. Ilmanlaadun suureita pystytään seuraamaan paikallisesti sekä internetin välityksellä. Energiansäästöä syntyy ja ilmanlaatu paranee ilmastointilaitteen automaattisen ohjauksen ansiosta.

Lähteet

1. Sisäilmayhdistys ry. Fysikaaliset tekijät. 2014. [Viitattu 19.2.2014] Saatavissa: <http://www.sisailmayhdistys.fi/terveelliset-tilat-tietojarjestelma/sisailmasto/fysikaaliset-tekijat/>
2. Allergia- ja Astmaliitto ry ja Hengityслиitto ry. Sisäilmaopas. [Viitattu 19.2.2014] Saatavissa: http://allergia-fi-bin.directo.fi/@Bin/537aa42007b95ea88d90bf2286b5d874/1392637649/application/pdf/734312/HeliSis%C3%A4ilmaopas_web.pdf
3. Seeed Wiki. CAN-BUS Shield. Päivitetty 2.4.2014. [Viitattu 13.4.2014] Saatavissa: http://www.seeedstudio.com/wiki/CAN-BUS_Shield
4. Seuri, Markku & Palomäki, Eero. Haasteellinen sisäilma: Riskianalyysi sisäilmaongelmissa. Tampere: Rakennustieto Oy. 2000. ISBN: 951-682-617-2
5. Aosong(Guangzhou) Electronics Co.,Ltd. DHT11 Product Manual. [Viitattu 2.4.2014] Saatavissa: <http://akizukidenshi.com/download/ds/aosong/DHT11.pdf>
6. Pulli,Tomi. Anturit ja Arduino. Aalto-yliopisto. 2014. [Viitattu 19.3.2014] Saatavissa: https://noppa.aalto.fi/noppa/kurssi/elec-a4010/luennot/ELEC-A4010_anturiluento.pdf
7. Arduino. 2014 [Viitattu 26.2.2014] Saatavissa: <http://arduino.cc/>
8. Salonen, Raimo O & Pennanen, Raimo. 2006. Pienhiukkasten vaikutus terveyteen. Tekes. [Viitattu 18.2.2014] Saatavissa: http://www.tekes.fi/Julkaisut/fine_terveys.pdf
9. Helsingin yliopisto. Sisäilman hiukkaslähteet. 2014. [Viitattu 18.2.2014] Saatavissa: <http://www.hiukkastieto.fi/node/201>
10. Koskinen, Jari. Mikrotietokonetekniikka Sulautetut järjestelmät. Helsinki: Otavan kirjapaino Oy.1998. ISBN: 951-1-14268-2
11. Sparkfun Electronics. RS485 Shield. 2014. [Viitattu 13.4.2014] Saatavissa: <https://www.sparkfun.com/products/11959>
12. Doukas, Charalampos 2012. Building Internet Of Things with the Arduino. ISBN: 1470023431.
13. Hengityслиitto ry. Ilmanpuhdistimet. 2014. [Viitattu 19.3.2014] Saatavissa: <http://www.hengityслиitto.fi/hengityслиlma/sisailma/ilmanpuhdistimet#.UylaBYXJdel>
14. Nardella, Davide. Arduino™ Ethernet communication library for S7 Siemens™ PLC. Päivitetty 20.11.2013. [Viitattu 20.3.2014] Saatavissa: <http://settimino.sourceforge.net/>
15. Helsingin teknillinen korkeakoulu. Analogia-digitaalimuunnos. 2003. Päivitetty 4.8.2003. [Viitattu 21.3.2014] Saatavissa: <http://signal.hut.fi/digis/luento1/admuunnos.html>
16. Sharp Corporation. GP2Y1010AU0F Compact Optical Dust Sensor. 2006. [Viitattu 31.3.2014] Saatavissa: http://sharp-world.com/products/device/lineup/data/pdf/datasheet/gp2y1010au_e.pdf
17. Sharp Corporation. Application note of Sharp dust sensor GP2Y1010AU0F. 2013. [Viitattu 31.3.2014] Saatavissa: http://sharp-world.com/products/device/lineup/data/pdf/datasheet/gp2y1010au_appl_e.pdf

18. Shirriff, Ken. Secrets of Arduino PWM. 2009. [Viitattu 9.4.2014] Saatavissa: <http://www.righ.to.com/2009/07/secrets-of-arduino-pwm.html>
19. Maxim Integrated Products, Inc. DS1307 Datasheet. 2008. [Viitattu 9.4.2014] Saatavissa: <http://datasheets.maximintegrated.com/en/ds/DS1307.pdf>
20. Sähköinfo Oy. ST 11.50 Suomen rakentamismääräyskokoelma D2. Rakennusten sisäilmasto ja ilmanvaihto. Määräykset ja ohjeet 2012
21. Nimim. ProvideYourOwn. Analog Output - Convert PWM to Voltage. Autodesk, Inc 2014. [Viitattu 14.4.2014] Saatavissa: <http://www.instructables.com/id/Analog-Output-Convert-PWM-to-Voltage/>
22. Wiznet Co., Inc. W5100 Datasheet Version 1.1.6. 2008.[Viitattu 14.4.2014] Saatavissa: https://www.sparkfun.com/datasheets/DevTools/Arduino/W5100_Datasheet_v1_1_6.pdf
23. Suomen automaatioseura ry. Rakennusautomaation hyödyt. 2005. [Viitattu 15.4.2014] Saatavissa: http://www.automatioseura.fi/index/tiedostot/BAFF_%20hyodyt.pdf
24. Microchip Technology Inc. PIC16C54C Datasheet. 1998. [Viitattu 22.4.2014] Saatavissa: <http://pdf1.alldatasheet.com/datasheet-pdf/view/74954/MICROCHIP/PIC16C54C.html>
25. Meptek Oy. Ilto 440 Suunnittelu-, asennus- ja käyttöohje. 2005.
26. LogMeIn, Inc. What is Xively. 2014. [Viitattu 24.4.2014] Saatavissa: http://xively.com/whats_xively/
27. Arduino. ModbusMaster Library for Arduino. 2014. [Viitattu 7.5.2014] Saatavissa: <http://playground.arduino.cc/Code/ModbusMaster>
28. Sisäilmayhdistys ry. Homevaurioiden ehkäisy ja tunnistaminen. 2014. [Viitattu 20.2.2014] Saatavissa: <http://www.sisailmayhdistys.fi/paasivuista-toinen/homevaurioiden-ehkaisy-ja-tunnistaminen/>

Laitteen piirikaavio

